



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JUSSI LEHTONEN
KIINTEISTÖAUTOMAATION TUOTTEISTUS JA TUOTERUNKOARKKI-
TEHTUURI

Diplomityö

Tarkastajat: prof. Matti Vilkkonen ja TkT
David Hästbacka
Tarkastaja ja aihe hyväksytty
5. lokakuuta 2016

TIIVISTELMÄ

JUSSI LEHTONEN: Kiinteistöautomaation tuotteistus ja tuoterunkoarkkitehtuuri
Tampereen teknillinen yliopisto
Diplomityö, 81 sivua, 5 liitesivua
Toukokuu 2017
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: prosessien hallinta
Tarkastaja: professori Matti Vilkkö, TkT David Hästbacka

Avainsanat: kiinteistöautomaatio, rakennusautomaatio, tuotteistus, tuoterunkoarkkitehtuuri

Uuden tuotteen luominen on yritykselle aina riskialtis projekti. Vaarana on taloudelliset tappiot ja yrityksen toimintafilosofiaan sopimattoman tuotteen suunnitteleminen. Tuotteistaminen täytyy ajatella prosessina, jossa hyödynnetään jatkuvaa arviointia ja kehityksen valvontaa. Kiinteistöautomaatiojärjestelmän tuotteistamisessa täytyy ottaa huomioon millainen on vallitseva markkinatilanne ja miten markkinoille on mahdollista päästä mukaan. Markkinatutkimus on tässä työssä suoritettu haastatteleamalla asiantuntijoita ja tutkimalla millaisia tuotteita kilpailevilla yrityksillä on.

Tuotteistamisprosessi on tässä työssä toteutettu innovointiprosessimallin avulla. Malli jakaa tuotekehityksen vaiheisiin ja portteihin. Vaiheissa tapahtuu suorittava työ ja portteilla arvioidaan vaiheissa tehtyjä tuotoksia.

Automaatio vaatii toimiakseen ohjelmiston, joten myös kiinteistöautomaatiojärjestelmän tuotteistaminen on ohjelmistotuotannollinen ongelma. Tehtävänä on mahdollisimman pienellä vaivalla, pystyä tuottamaan asiakkaille räätälöityjä tuotteita.

Tuotannolliseen ongelmaan vastataan hyödyntämällä kiinteistöautomaationjärjestelmän kehittämisessä tuoterunkoarkkitehtuuria, jonka avulla saadaan tulevaisuudessa nopeasti kehitettyä ratkaisuja asiakkaille. Tuoterunkoarkkitehtuurien suurimmat ongelmat ovat tuoterungon muutosten hallinta sekä kokonaisuuden hahmottaminen hallinta. Muutoshallinnan ongelmaan on työssä vastattu rakentamalla tuoterunkoarkkitehtuuri modulaarisista black-box-komponenteista, joita on mahdollista käyttää joustavasti tuoteperheen eri tuotteissa. Kokonaisuuden hahmottamista ja hallintaa on pyritty helpottamaan suunnittelemalla ohjelmointisäännöt ja dokumentoimalla ohjelmointikoodi.

Työn tuloksena saatiin tuotekehitettyä kiinteistöautomaatiojärjestelmälle tuoterunkoarkkitehtuuri. Kehitetty tuote vaatii kuitenkin vielä jatkokehitystä. Erityisesti tuotteen testaaminen ja tuotteistusprosessi täytyy suorittaa loppuun.

ABSTRACT

JUSSI LEHTONEN: Building automation system's productization and product line architecture

Tampere University of Technology

Master of Science Thesis, 81 pages, 5 Appendix pages

May 2017

Master's Degree Programme in Automation Technology

Major: Process control

Examiner: Professor Matti Vilkkö, Postdoctoral Researcher David Hästbacka

Keywords: building automation, productization, product line architecture

Making a new product is always a riskful project for any company. The dangers are financial losses and making a product that doesn't fit in the company's operational philosophy. Productization must be thought as a process, where continuous evaluation and supervision of development are being utilized. When one is productizing a building automation system, one must look into what is the prevalent market situation and how can the company get a share of this market. In this thesis a market research is made by using theme interviews of experts and inspecting what the competitive products have to offer.

In this thesis the product, product line architecture of building automation system, is developed by using an innovation process model, which separates the development into phases and gates. Phases mean different work phases. Gates are there for evaluating the phases' outputs.

In order to function an automation system needs a software so when a building automation system is being productized it's also a software engineering problem. The objective is to deliver solutions to clients as effectively as possible.

The answer to the problem is to utilize a product line architecture in building the automation system. With a product line architecture, it is possible to generate products to clients very effectively. Problems with the product line architecture are to control modifications between different versions of the program and to manage the whole program. In this thesis the modification problem is being answered by building a product line architecture with modular black-box-components which can be used flexibly within the software and product family. To help to manage the product line, the software is documented and programming rules are made.

The result of the thesis is product line architecture for a building automation system. The developed product still needs some further development. Especially testing needs more attention.

ALKUSANAT

Tämän työn tekeminen oli vaativaa ja haastavaa, mutta antoisaa. Tahdon kiittää Petri Pintaa ja Jari Mäkelää mielenkiintoisesta aiheesta. Petriä tahdon kiittää myös erinomaisesta ohjauksesta ja innostamisesta työn tekemisen aikana. Kiitän työkavereitani Sakua ja Jarno työpäivien keventämisestä ja avusta, jota sain moniin ongelmiin. Lisäksi kiitos perheelleni ja ystävilleni työn teon aikana saamastani tuesta. Erityisesti tahdon kiittää Veeraa, kaikesta.

Tampereella 24.5.2017

Sisältö

1	Johdanto	1
1.1	Mitä on kiinteistöautomaatio?	1
1.2	Kiinteistöautomaation kehittyminen	2
1.3	Kiinteistöautomaation ohjelmistosuunnittelun kehittäminen.....	6
2	Uuden tuotteen kehittäminen	8
2.1	Miksi tuote epäonnistuu tai onnistuu?.....	8
2.2	Stage-Gate-innovointiprosessimalli	12
3	Tuoterunkoarkkitehtuurit	16
3.1	Tuoterunkoarkkitehtuurin tuotantoprosessi	19
3.2	Tuoterunkoarkkitehtuurien jatkuva kehittäminen	22
3.3	Tuoterunkoarkkitehtuurin ylläpito	25
3.4	Tuoterunkojen ongelmat	26
4	Kiinteistöautomaatiojärjestelmän tuotteistus	29
4.1	Kiinteistöautomaation taustatutkimus.....	29
4.2	Markkinatutkimus	37
4.3	Toimialakohtainen haastattelututkimus kiinteistöautomaatiosta	40
4.4	Kiinteistöautomaatiojärjestelmän tavoitteet.....	45
4.5	Tuotekehityksen organisointi ja komponentit.....	46
5	Kiinteistöautomaatiojärjestelmän tuoterunkoarkkitehtuurin kehitys	49
5.1	Tuoterunkoarkkitehtuurin esitutkimus	49
5.2	Tuoterunkoarkkitehtuurin vaatimukset	51
5.3	Tuoterunkoarkkitehtuurin arkkitehtuurityypin valinta.....	55
5.4	Tuoterunkoarkkitehtuurin toteutusvaihe	58
5.5	Tuoterunkoarkkitehtuurin luokkakaavio	70
5.6	Tuoterunkoarkkitehtuurien testaaminen.....	70
5.7	Dokumentointi	72
6	Tulosten arviointi	73
7	Yhteenveto	77
8	Lähdeluettelo.....	79

LYHENTEET JA MERKINNÄT

Black-box-komponentti	Ohjelmointikomponentti, jonka sisäisestä koodista ei ohjelmoijalla tarvitse olla tietoa
EU	Euroopan Unioni
FBD	Function block diagram -ohjelmointikieli
IND	Tilatieto
IO	Input-Output
LS	Lämmönsiirtojärjestelmä
LVI	Lämpö-vesi-ilma
MAIN(PRG)	Pääohjelma TwinCat2 -ohjelmistossa
PID	Proportional-integral-derivative, säädintyyppi
PLC	Programmable logic controller, ohjelmoitava logiikka
PK	Poistoilmakone
PRG	Aliohjelma TwinCat2 -ohjelmistossa
SJ	Sähköjärjestelmä
ST	Structured text -ohjelmointikieli
TK	Tuloilmakone
Q	Ulostulo

1 JOHDANTO

Nykyihminen viettää elämästään suurimman osan sisätiloissa. Jotta elämä sisätilassa olisi mahdollisimman mukavaa, turvallista ja taloudellista, ihmisen on aktiivisesti pyrittävä vaikuttamaan sisätilan olosuhteisiin. Menneisyydessä näistä asioista huolehtiminen jätettiin talonmiehille. Nykyään koneet ovat ilmestyneet rakennuksiin ja syrjäyttäneet talonmiehet. Kiinteistöautomaatio on teknologiaa, jonka olemassa oloa ei asukas ideaalitilanteessa edes huomaa. Rakennusautomaation ja LVI (lämpö-vesi-ilma)-tekniikan kehittyminen on osa yhteiskunnan teknologista kehitystä, jolla on sekä infrastruktuurille että rakennusten käyttäjille oma merkityksensä. Automaation mahdollisuudet kiinteistöjen olosuhteiden ja toimintojen ohjaamisessa ovat laajat.

Suomella ja Euroopan Unionilla on kovat tulevaisuuden tavoitteet kasvihuonepäästöjen vähentämiseksi. Eurooppalaisissa rakennuksissa kulutetaan enemmän energiaa, kuin mitä liikenne ja teollisuus kuluttavat yhteensä. Suomen ja Euroopan Unionin energiansäästötoimenpideohjelman mukaan vuonna 2020 energiansäästötavoite on 20 % vuoden 2012 tasosta. Energiankulutusta olisi mahdollista vähentää jopa viidenneksellä rakennusten energiatehokkuutta parantamalla. [1, p. 1–3] Kiinteistöautomaation mahdollisuudet eivät siis rajoitu ainoastaan olosuhteiden hallintaan. Niillä voi olla konkreettisia hyötyjä ilmaston ja talouden kannalta. Energiatehokkuus on mukavuuden ja turvallisuuden ohella merkittävin asia, johon rakennusautomaatiolla voidaan positiivisessa mielessä vaikuttaa. Automaation oikeanlaisella lisäämisellä voidaan edistää energiansäästötavoitteiden saavuttamista. Energiatehokkaiden rakennusten rakennuttaminen kuuluu myös olennaisesti Euroopan Unionin energiansäästöpolitiikkaan. [1, p. 1–3] Joh-tuen hallinnollisesta painostuksesta ja ihmisten yleisestä säästämishalusta rakennusau-tomaatiojärjestelmille on tulevaisuudessa kysyntää. Kiinteistöautomaatiolla tulee ole-maan tulevaisuudessa suuri merkitys suomalaisessa rakennusteollisuudessa.

1.1 Mitä on kiinteistöautomaatio?

Kiinteistöautomaatio on joukko yksittäisiä tai verkotettuja laitteita, joilla on erilaisia tavoitteita kiinteistön toiminnan suhteen. Yksinkertaistettuna kiinteistöautomaation tavoitteet voidaan jakaa neljään osioon, jotka ovat vuorovaikutuksessa keskenään. Tavoitteet ovat turvallisuuden, kiinteistönkäytön mukavuuden, optimaalisen energian kulutuksen ja kiinteistön käytön taloudellisuuden parantaminen. Turvallisuuskysymykseen voidaan vastata ottamalla automaation hallintaan esimerkiksi tulipalon leviämisen eston automatiikkaa. Mukavuuden lisäämiseen taas liittyy arkisten – usein toistuvien – askareiden automatisointi, kuten automaattinen valojenohjausjärjestelmä. Energian hallintaan taas liittyy lämmityksen ja jäähdytyksen säätely ja tietysti myös automaattinen va-

laistuksenhallinta. Edellisten asioiden optimoinnilla taas saadaan aikaan kiinteistön käytön säästöjä. Tavoitteiden painotus riippuu paljon siitä, millaisessa kohteessa kiinteistöautomaatiojärjestelmä on. Pienessä asunnossa painotus on hyvin todennäköisesti asumismukavuudessa. Tällaisessa kohteessa kiinteistöautomaatiosta puhutaan usein kotiautomaationa. [2, p. 18] Asumismukavuuteen voidaan panostaa esimerkiksi valaistuksenohjauksia kehittämällä ja automatisoimalla usein toistuvia toimintoja. Myös energian- ja lämmityksen hallinta on tärkeä osa asumismukavuutta ja yksityisten asuntojen automaatiojärjestelmiä. [3, p. 2–3] Suuremmissa kohteissa kuten toimistorakennuksissa painotus on usein järjestelmän käytön tehokkuudessa: energiatehokkuudessa ja energiankulutuksen optimoinnissa. [2, p. 18] Optimoinnilla voidaan tarkoittaa esimerkiksi toimistorakennuksen sisälämpötilan hallintaa. Liian korkea tai matala lämpötila vaikuttaa negatiivisesti työtehoon. [3, p. 2–3] Yhteistä kaikille kiinteistöautomaatiojärjestelmille on, etteivät järjestelmän käyttäjät todennäköisesti ole alan asiantuntijoita, vaan järjestelmän hallinta on jätetty koulutetuille huoltomiehille. [2, p. 18] Erityisesti tämä korostuu kotiautomaation kohdalla. [4, p. 9] Eri kiinteistöautomaatiojärjestelmien suurin ero on kuitenkin käytännössä niiden kokoluokka. Ohjattavat ja hallittavat laitteet ja kohteet ovat hyvin samankaltaisia kokoluokasta riippumatta. Kun kiinteistöautomaatiojärjestelmä toimii oikein, käyttäjä ei kiinnitä siihen mitään huomiota.

Automaation mahdollisuudet kiinteistöjen järjestelmien ohjaamisessa ovat hyvin laajat. Asiaa ei yksityiskohtaisemmin tässä työssä käydä, sillä aiheesta on tehty useita opinnäytetöitä. Esimerkiksi Joni Helaakoski: *Kotiautomaation mahdollisuudet* (Centria, 2011) ja Annina Rönnkvist: *Kiinteistöautomaatiosovellukset* (Centria, 2010).

1.2 Kiinteistöautomaation kehittyminen

Kiinteistöautomaatiojärjestelmien kehittyminen on tapahtunut yhteiskunnan ja teknologian kehityksen mukana. Kuvassa 1.1 on esitettyä kiinteistöautomaatiojärjestelmien evoluutio hierarkkisena mallina. Ylhäällä on nykyaika ja alempana menneisyys.

Älytalo
Verkottunut järjestelmä
Integroitu järjestelmä
Yksittäiset laitteet

Kuva 1.1 Kiinteistöautomaatiojärjestelmien evoluutio. Ylimpänä nykyaika ja alimpana menneisyys. Mukailtu lähteestä [15, p. 19].

Evoluution ensimmäisenä asteena ovat yksittäiset laitteet, jotka toimivat itsenäisesti. Yleensä niissä on sisäinen anturi ja toimilaite. Termostaatti on hyvä esimerkki tällaisesta laitteesta. Toisena asteena on integroitu järjestelmä, joissa yksittäisiä laitteita ohjataan releohjauksilla. Tällä tasolla yksittäiset laitteet on yhdistetty yksinkertaiseksi automaatiojärjestelmäksi. Kolmas evoluution aste on verkottuneet järjestelmät, jotka sisältävät paljon enemmän vuorovaikutuksessa olevia kokonaisuuksia kuin evoluution edellinen aste. Verkottuneissa järjestelmissä on hierarkkisesti säätöalgoritmeilla ohjattuja koko-

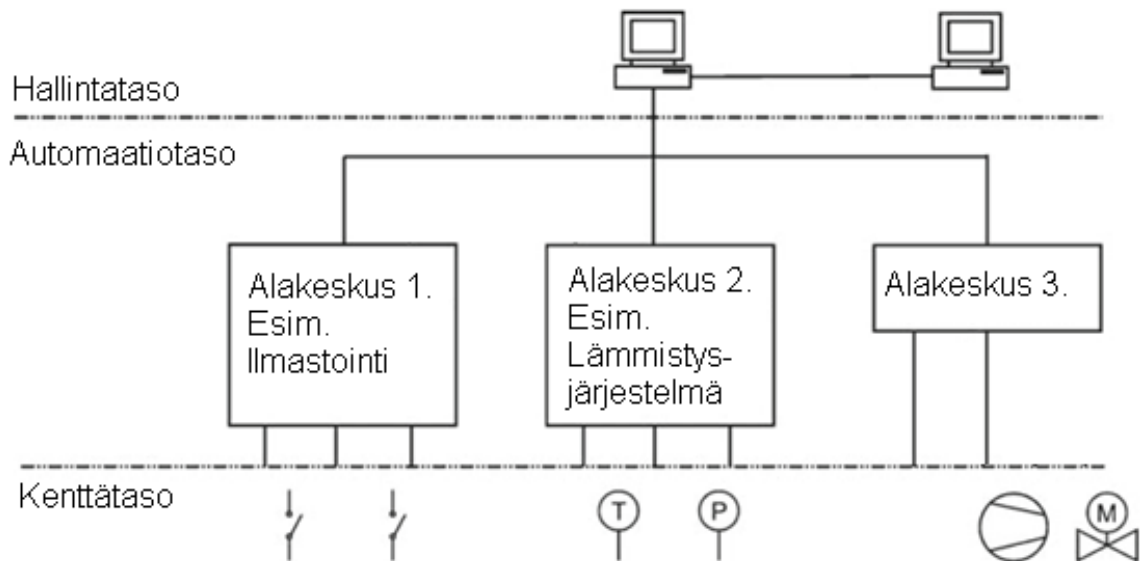
naisuuksia, jotka ovat vuorovaikutuksessa keskenään. Ilmanvaihtokoneet, lämmitysjärjestelmät ja muut kiinteistön ohjausjärjestelmät keskustelevat keskenään. Suurin osa nykyaikaisista kiinteistöautomaatiojärjestelmistä on tällä evoluution asteella. Älytalo on evoluution ylin aste ja tulevaisuuden tavoite. [15, p. 19] Älytalolle on olemassa useita eri määritelmiä, mutta käytännössä älytalon on tarkoitus kattaa neljä päätekijää, jotka on esitetty taulukossa 1.1.

Taulukko 1.1: *Älytalojärjestelmän päätekijät.* [5]

Rakenteet ja osat	Talotekniset järjestelmät	Palvelut	Hallinnointi
<ul style="list-style-type: none"> – Ulkovaipan erityiskyky – Materiaalien vaihdettavuus 	<ul style="list-style-type: none"> – Lämmitys – Ilmanvaihto – Sisäilmaolosuhteet – Valaistus – Tietoliikenne – Ohjaukset – Valvonta 	<ul style="list-style-type: none"> – Toimistoautomaatio – Ääni-, data-, ja videoviestintä – Turvallisuuden hallinta. 	<ul style="list-style-type: none"> – Käytön ja kunnossapidon hallinta – Energiatohokkuusraportit – Kustannustenhallinta – Kehityssuunta-analyysit – Informaation hallinta.

Älytalo on siis kattava järjestelmä, joka koskettaa kaikkia asumisen ja kiinteistön käytön osa-alueita rakennuksen rakenteista aina rakennuksen hallintaan sekä rakennuksen tuottamiin palveluihin. Kaiken taustalla on, että rakennuksen järjestelmien tulee olla vuorovaikutuksessa ihmisen kanssa – ei pelkästään muiden koneiden kanssa. Tarkoituksena on tarjota käyttäjälle palveluita, mukavuutta ja informaatiota. Älytalo ei ole käyttäjästä irrallinen kokonaisuus, vaan sen on tarkoitus oppia ja reagoida asukkaan toimintaan. [17]

Automaatiojärjestelmien tulee olla ohjattavissa koordinoitusti, jotta verkottunutta järjestelmää voidaan parhaiten hyödyntää. Näin saavutetaan mahdollisimman suuret ohjattavuus- ja energiansäästöedut. Jotta verkottunutta järjestelmää voidaan ohjata, täytyy ymmärtää, miten automaatiojärjestelmät koostuvat. Useimmiten kiinteistöautomaatiojärjestelmät rakennetaan hierarkkisenä järjestelmänä, joka määritellään yleensä kolmeen eri tasoon: hallinta-, automaatio- ja kentätasoon. Kuvassa 1.2 on havainnollistettu kiinteistöautomaatiojärjestelmän rakenne.



Kuva 1.2 Kiinteistöautomaatiojärjestelmän hierarkkinen rakenne. Ylätason säädöstä siirrytään aina kohti alemmaa tasoa, jolloin saavutetaan optimoitu säätö. Mukailtu lähteestä [3, p. 7]

Hallintatasolla ovat automaatiojärjestelmän valvonta- ja hallintalaitteistot kuten paikallis- tai keskusvalvomot. Näillä laitteilla ohjataan koko kiinteistöautomaatiojärjestelmän toimintaa. Valvomoita voi olla useita, eivätkä ne välttämättä ole ohjattavassa rakennuksessa. Tietoliikennetekniikan ansiosta on hyvin tavallista, että kiinteistöautomaatiojärjestelmän käyttäjärajapintaan pääsee käsiksi myös etäkäyttäjänä Internet-yhteydellä. Ohjaaminen voidaan siis tehdä vaikka toiselta puolen maailmaa. Nykyään käyttäjärajapintana toimii käytännössä aina graafinen käyttöliittymä, jolla prosessin toimintaa voidaan sekä seurata että hallita. Samoin rajapinnan kautta hoituu järjestelmän raportointi hälytyksistä, virhetilanteista ja muista oleellisista tapahtumista.

Automaatiotasolla ovat automaatiojärjestelmän alakeskukset eli ohjausjärjestelmät, joita ohjataan valvomoista käsin. Alakeskuksessa on kunkin automaatiojärjestelmän ohjauslogiikka, joka ohjaa toimilaitteita antureilta saadun informaation ja ohjauslaitteelle ohjelmoitujen periaatteiden mukaisesti. Yhdessä kiinteistöautomaatiojärjestelmässä voi olla useita alakeskuksia, jotka toimivat koordinoitusti. [6, p. 93–94] Nykyään ohjaustietokoneena toimii usein ohjelmoitava logiikka (PLC). Järjestelmän PLC:t toimivat alakeskuksien aivoina: niissä tapahtuu laskenta-, säätö- ja muut toimintaprosessit.

Säätimen tehtävänä on pitää säädettävä suure asetetussa tai esimääritetyssä arvossa tai määritettyjen rajojen sisällä häiriöstä huolimatta. Suljettu säätöpiiri pyrkii pitämään erosuureen mahdollisimman pienenä. Esimerkki häiriöstä järjestelmässä on ulkolämpötila, joka vaikuttaa häiriönä sisäilman lämpötilaan. Kiinteistöautomaatiojärjestelmien säätimien ei tarvitse olla viritetty kovin nopeiksi, koska olosuhteiden muutokset ovat rakennuksissa hitaita johtuen suurista tiloista. Myöskään pienet säätövirheet esimerkiksi lämpötilan säädössä eivät ole kovin vakavia. Nopeuden sijaan säädössä tärkeintä on toimintavarmuus: säädön pettäminen aiheuttaa potentiaalisesti kiinteistön käyttäjille paljon haittaa joko kustannuksina tai olosuhteiden epämukavuutena. [6, p. 55–56]

Yleensä kiinteistöautomaatiojärjestelmiä säädetään erillisinä silmukoina, hyödyntäen kaskadisäätöä tai portaittaista säätöä. Koska suurimpana vaatimuksena järjestelmissä on toimintavarmuus, ja säätötavat eivät ole kovin monimutkaisia. Yleisin kiinteistöautomaatiojärjestelmien säätöalgoritmi on PID-säädin (proportional-integral-derivative-säädin). [6, p. 55–60] Kiinteistöautomaatiojärjestelmiä säädetään myös paljon säätökäyrien avulla, eli asetusarvo muuttuu suhteessa johonkin muuttujaan. Esimerkiksi rakennuksen lämmitysveden lämpötila muuttuu suhteessa ulkolämpötilaan. Nykyään kiinteistä säätöarvoista on siirrytty yhä enemmän tosiaikaiseen säätöön. Hyvä esimerkki tästä on läsnäoloon perustuva säätö hotellihuoneissa: ilmanvaihto on pois päältä ja lämmitys pienemmällä tasolla, kun huone on tyhjillään. Lisäksi yleinen tapa säätää järjestelmiä on kalenteriin pohjautuva päälle/pois-säätö. Kalenteriin voidaan asettaa milloin lämmitetään ja milloin jäähdytetään. Esimerkiksi yöllä tai työpäivän jälkeen toimistorakennuksen ilmanvaihto ei ole päällä. [3, p. 13–15]

Automaation hierarkian alimmalla tasolla eli kenttätasolla on kiinteistöautomaatiojärjestelmän toimilaitteet ja anturit. Kenttätason laitteet toimivat alakeskuksen tiedon vastaanottajana, välittäjänä ja lähettäjänä. Käytännössä ne toteuttavat valvomosta alakeskukselle syötetyn käskyn. [6, p. 93–94] Anturointi on kiinteistöautomaatiojärjestelmän tärkeimpiä komponentteja – ilman sitä ei takaisinkytketty säätö voi toimia. Anturoinnin avulla järjestelmästä saadaan PLC:lle tarvittava informaatio säädön toteuttamiseen. Kiinteistöautomaatiojärjestelmien kehittyessä, anturien määrä järjestelmissä on samalla kasvanut. [4, p. 12]

Toimivan säädön takaamiseksi on järjestelmän tasojen rajapintojen oltava hyvin määriteltyjä. Kommunikaatio tasojen välillä tapahtuu kenttäväylien avulla. Aiemmin kenttäväylät olivat käytännössä aina valmistajien itsensä suunnitteleamia, mutta nykyään avoimien protokollien kenttäväylät ovat yleistyneet. Suurin syy muutokseen on ollut järjestelmien muuttuminen yhä monimutkaisemmiksi ja integraatiovaatimusten kasvaminen. Tämä on kasvattanut kenttäväyliin kohdistuvia vaatimuksia, eikä kenttäväylien yksilöllinen, yrityskohtainen kehittäminen, ole enää ollut kannattavaa. Avointen kenttäväylätekniikoiden yleistyminen on vähentänyt kiinteistöautomaatiojärjestelmien riippuvuutta yksittäisistä toimijoista: eri valmistajien komponentit ovat yhä useammin toistensa kanssa yhteensopivia. Jos valmiiseen järjestelmään tehdään muutoksia, ei uusia komponentteja välttämättä tarvitse tilata alkuperäiseltä toimittajalta, mikä on lisännyt kilpailua alalla. Kiinteistöautomaatiojärjestelmissä voidaan käyttää samanaikaisesti useita eri kenttäväyliä. Jokaisella kenttäväylätyypillä on omat etunsa ja puutteensa. [3, p. 20–22]

Kiinteistöautomaatiojärjestelmien toimintaikä on noin 10–15 vuotta. Tämän jälkeen automaatiojärjestelmän vaatimien vara-osien saatavuus alkaa olla heikkoa. Toisin sanoen toimilaitteiden kestävyys kertoo sen, kuinka pitkäikäisiä kiinteistöautomaatiojärjestelmät ovat. Ohjelmointi ja säätöalgoritmit eivät sinänsä vanhene. [7]

Ohjelmistosuunnittelun suurin ongelma kiinteistöautomaatiojärjestelmissä on suunnittelun organisointi tehokkaasti. Perinteinen tapa ohjelmistojen tekemisessä on kehittää järjestelmiä yksi kerrallaan kopioiden yhteisiä ominaisuuksia. Ohjelmistotuotannon tehokkuutta voidaan lisätä suunnittelemalla ohjelmapohja, jonka päälle uusia

projekteja voidaan rakentaa. Tästä seuraava kysymys on, mitä asioita ohjelmapohjaan liitetään. Kiinteistöautomaatioprojektit voivat näyttää samanlaisilta: vierekkäiset talot eivät välttämättä ulkoisesti eroa toisistaan. Kuitenkin talojen välillä voi ohjelmatasolla olla paljonkin vaihtelevuutta. Lisäksi eri asiakkailta on erilaisia vaatimuksia.

1.3 Kiinteistöautomaation ohjelmistosuunnittelun kehittäminen

Työn tavoitteena on kehittää kiinteistöautomaatiojärjestelmää ohjaavalle sovellukselle ohjelmapohja, jota voidaan hyödyntää tulevaisuuden projekteissa. Ohjelmapohjan tarkoituksena on nopeuttaa projektien läpivientiä ja siten alentaa projektiin kohdistuvia kustannuksia. Päättökysymys on, millainen ohjelmapohja kiinteistöautomaatiojärjestelmälle kannattaa toteuttaa? Tämän kysymyksen selvittämiseksi täytyy vastata seuraaviin alakysymyksiin. Ensimmäinen on, kannattaako kiinteistöautomaatiojärjestelmälle kehittää ohjelmapohjamuotoinen ratkaisu? Kiinteistöautomaatiojärjestelmiä on eri kokoisia ja erilaisiin käyttötarkoituksiin. Jos kiinteistöautomaatiojärjestelmissä on liian paljon keskinäistä vaihtuvuutta, ei ohjelmapohjamuotoinen ratkaisu todennäköisesti vastaa kunnolla erilaisuuden aiheuttamiin haasteisiin. Jos ohjelmapohja kannattaa kehittää, niin millainen tasapaino ohjelman kiinteiden ja vaihtuvien ominaisuuksien välille saadaan. Toinen alakysymys on, miten toteutetaan ohjelmapohjan kehitysprosessi. Prosessin täytyy olla organisoitu ja hallittu, jotta kehitysprosessi olisi mahdollisimman tehokas.

Ohjelmapohja kehitetään tekemällä ohjelmapohjasta tuoterunkoarkkitehtuuri, jolla voidaan hallita tuoteperheitä ja niiden kehitystä. Tuoterunkoarkkitehtuurin tarkoituksena on toimia tuoteperheen tuotteiden kehityspohjana, eli tuoterunkoarkkitehtuurissa on pysyviä osia, mutta myös muuttuvia osia. Hallitun tuotekehitysprosessin toteuttamiseen käytetään State-Gate-innovointiprosessimallia, joka jaksottaa tuotekehityksen vaiheisiin ja vaiheita arvioiviin portteihin.

Suuri ongelma tutkimuksessa on selvittää, millainen on hyvä kiinteistöautomaatiojärjestelmä. Osana innovointiprosessia työssä suoritetaan markkinatutkimus, jonka tarkoituksena on vastata millaisia kiinteistöautomaatiojärjestelmiä nyt on käytössä ja miten niitä olisi mahdollista kehittää paremmiksi. Markkinatutkimus toteutetaan laadullisen analyysin avulla: tutkimalla kirjallisuutta, kilpailijoiden markkinointimateriaaleja sekä haastattelemalla kiinteistöautomaatiosta vastaavia huoltomiehiä, valvomotyöntekijöitä ja yhtä kiinteistön omistajaa. Haastattelut toteutetaan avoimina teemahaastatteluin. Työn tarkoituksena ei ole siis ainoastaan tuoterunkoarkkitehtuurin kehittäminen, vaan myös kiinteistöautomaatiojärjestelmän tuotteistaminen.

Työn toisessa luvussa esitellään käytetty innovointiprosessimalli ja pohditaan, millainen on hyvä tuote. Kolmannessa luvussa esitellään tuoterunkoarkkitehtuuri. Neljännessä luvussa selvitetään markkinatutkimuksen avulla, millainen on hyvä kiinteistöautomaatiojärjestelmä. Viidennessä luvussa käydään läpi tuoterunkoarkkitehtuurin kehi-

tysprosessi. Kuudennessa luvussa käsitellään työssä saatuja tuloksia ja arvioidaan onnistumista.

2 UUDEN TUOTTEEN KEHITTÄMINEN

Yritykset ovat jatkuvassa kilpailutilanteessa toistensa kanssa, siten myös kehityksen on oltava jatkuvaa, jotta kilpailussa mukana pysyminen on mahdollista. Paikalleen jääminen voi tarkoittaa yrityksen näivettymistä. Yleinen kehittymiskeino on uusien tuotteiden innovointi. Uudet tuotteet ovat avain yrityksen menestymiseen, sillä onnistuessaan ne nostavat yrityksen arvoa tuomalla lisää asiakkaita, mainetta, teknologisen etulyöntiaseman ja tuloja. [8, p. 9] Tarkoitus on löytää tuote tai lähestymistapa, joka on parempi kuin kilpailijoilla. Lisäksi teknologian kehityksellä pyritään usein vastaamaan asiakkaiden muuttuneisiin tarpeisiin. Esimerkiksi mikroprosessorien nopea kehittyminen on tuonut sekä uusia teknologisia mahdollisuuksia että muuttanut monien asiakkaiden tarpeita ja vaatimuksia. Myös kilpailu itsessään on viime vuosikymmeninä kiristynyt, sillä globalisaatio on avannut maailmanlaajuiset markkinat yritysten ulottuville. Globalisaatio on kuitenkin kiristänyt myös sisämarkkinoiden kilpailutilannetta. Yritykset innovoivat pysyäkseen elinvoimaisina. [8, p. 15–17]

2.1 Miksi tuote epäonnistuu tai onnistuu?

Innovointi ei ole yksinkertainen prosessi. On vaikea kehittää täsmälleen sellainen tuote, jonka markkinat vaativat. Toisin sanoen innovointiin liittyy aina riskejä. Epäonnistuneisiin tuotteisiin kuluu tärkeitä resursseja: aikaa ja rahaa. [8, p. 9]

Hyvin yleinen syy tuotteen epäonnistumiseen on, ettei kehitetty tuote erotu riittävästi kilpailijoiden tuotteista. Samankaltaisuus ei houkuttele uusia asiakkaita kiinnostumaan tuotteesta. Jos erottautumista ei tapahdu, on suunnitteluprosessissa laiminlyöty riittävän taustatutkimuksen tekeminen. Innovointiprosessissa on tärkeää tietää, mitkä ovat kohdeasiakaskunnan tarpeet: mitä he tuotteelta tarvitsevat ja mitkä ovat heidän toiveensa uutta tuotetta kohtaan. Lisäksi on tunnettava kilpailevien tuotteiden ominaisuudet ja ymmärrettävä omat vahvuudet ja heikkoudet. On osattava katsoa markkinoita monesta eri näkökulmasta. [8, p. 27–28] Yrityksen tekemien tuotteiden täytyy olla yrityksen oman strategian mukaisia, jotta yritys pystyy kehittymään haluttuun suuntaan. Strategiasta on pääasiassa vastuussa yrityksen johto. [9, p. 5]

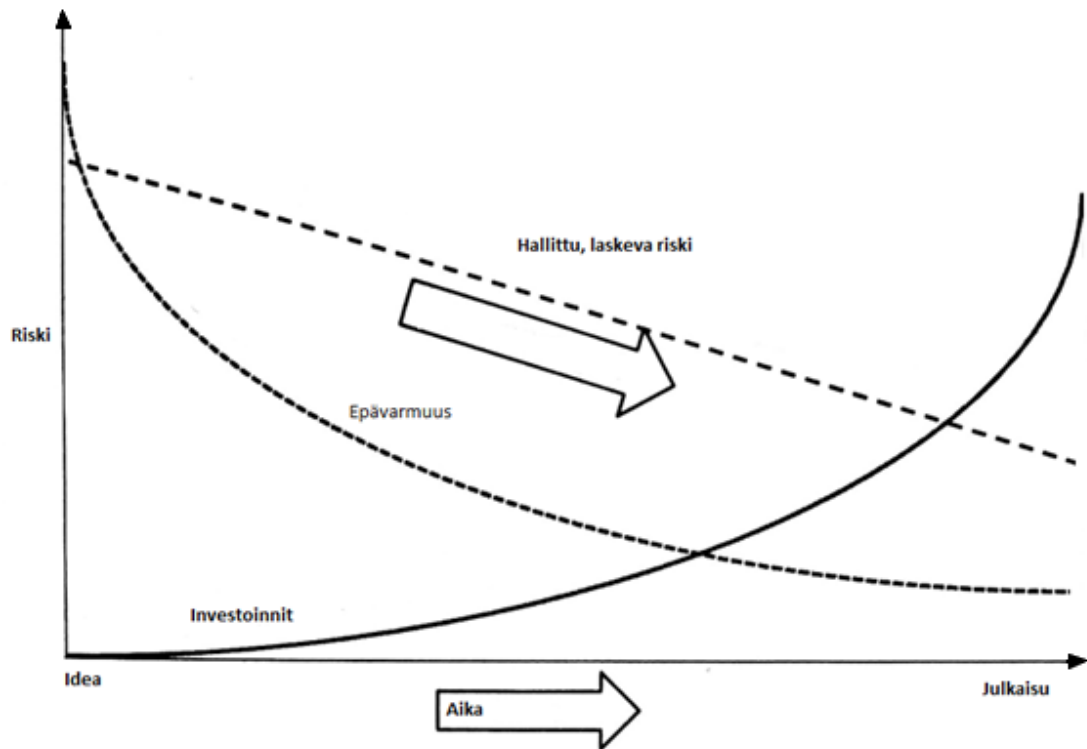
Toinen merkittävä syy epäonnistumiseen on epävakaa suunnitteluprosessi, josta edellä kuvattu näköalattomuus markkinoiden tuntemuksesta voi olla enne. Suunnitteluprosessin täytyy olla riittävän hyvin kontrolloitu, jotta suunniteltu tuote pysyy alkupe-
räisen suunnitelman mukaisena. Suunnittelun tärkeyttä on osattava kunnioittaa – kotimaan markkinoille suunniteltu tuote ei voi ilman perusteellista suunnittelutyötä laajentua koskettamaan globaaleja markkinoita. Heikosti toteutettu suunnitteluprosessi tarkoittaa usein myös projektitiimien tehottomuutta. Usein ongelmana on myös keskittyminen

liian moneen projektiin samaan aikaan. Tällöin työntekijät ovat ylityöllistettyjä, eikä yhteenkään projektiin saada tarpeen mukaista työpanosta. Ylityöllistäminen johtaa aikataulujen pettämiseen, mikä voi johtaa itseään vahvistavaan kierteseen. Myös seuraava projekti kärsii edellisen projektin pettäneestä aikataulusta. Yrityksen onkin pystyttävä tunnistamaan omat voimavaransa, eikä ryhtyä liian kunnianhimoisiin projekteihin. Jos projektin toteuttamiseen ei ole riittävää osaamista, tulos jää usein asetetuista tavoitteista. [8, p. 29–31]

Innovointiin liittyy siis monia mahdollisia kompastuskiviä, joiden vaikutus suunnitteluprosessin onnistumiseen on hyvin merkittävä. Yrityksen tuntemus sekä asiakkaidensa tarpeista että omista tarpeistaan ja kyvyistään on onnistumisen edellytys. Epäonnistuminen tietää rahallisia tappioita, mutta prosessin onnistuessa mahdollinen palkinto voi olla suuri. Melkein puolet tuotekehitysprojekteista ei saavuta niille ennalta määritellyjä tavoitteita [10, p. 46]. Onnistuneeseen innovointiin tarvitaan Cooperin mukaan neljää asiaa:

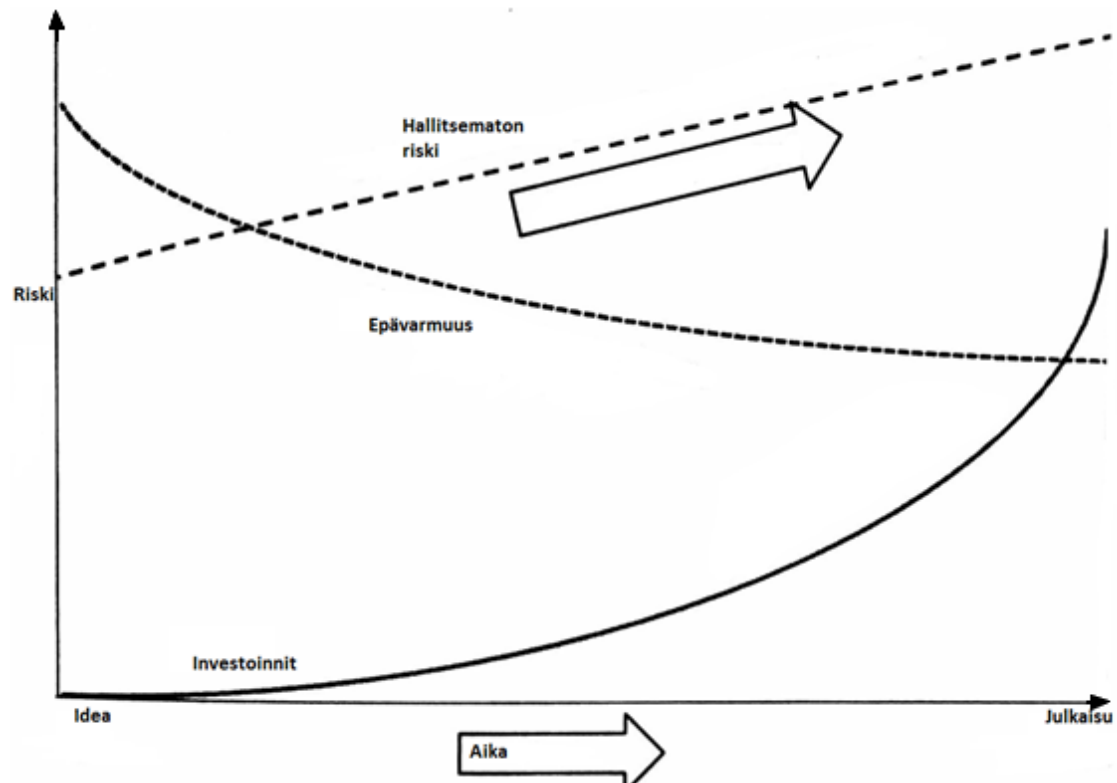
- teknologista osaamista
- riittävästi resursseja
- positiivista ilmapiiriä ja kulttuuria
- oikeanlainen innovointiprosessimalli, jolla poimitaan onnistuneet ajatukset. [8, p. 6]

Edellisten asioiden lisäksi, Cooper nostaa hyvin tärkeäksi osaksi onnistunutta suunnittelua myös nopeuden. Nykyajan muuttuvilla markkinoilla usein vahvimmissa on yritys, joka pystyy vastaamaan asiakkaiden muuttuneisiin tarpeisiin ensimmäisenä. [8, p. 32] Luonnollisesti nopeus ja huolellinen suunnittelu ovat usein ristiriidassa, mikä luo jo itsessään paineita innovoinnille. Oikeanlainen innovointiprosessimalli tarkoittaa suunnittelun organisointia ja kontrollointia, minkä uudelle tuotteelle tärkeät asiat onnistutaan löytämään. Kehitysmalli toimii myös riskinhallintakeinona. Se auttaa ymmärtämään, milloin otettava riski on liian suuri [8, p. 94–95]. Kuvassa 2.1 on esitetty, miten riskin hallinta vaikuttaa projektiin vaikuttavaan epävarmuuteen suunnitteluelinkaaren aikana.



Kuva 2.1 Riskin hallinnan vaikutus epävarmuuteen ja kokonaisriskiin. Kun riski on hallittu, se on alimmillaan projektin loppupisteessä. Mukailtu lähteestä [8, p. 97]

Kuvaajan pystyakseli kuvaa riskin suuruutta. Vaaka-akseli kuvaa projektin etenemistä. Alkupisteessä projekti on idean tasolla. Oikealla on valmiin tuotteen julkaisuhetki. Alkuvaiheessa projektiin liittyvä epävarmuus on hyvin suuri, mutta nopeasti laskeva. Alkuvaiheessa projektiin sijoitettu raha on luonnollisesti nolla, mutta mitä pidemmälle projekti etenee sitä enemmän ja kiihtyvämällä tahdilla projektiin sijoitetaan rahaa. Mitä enemmän projektissa on rahaa kiinni sitä miellyttävämpää on mitä alhaisempi riski siihen liittyy. Riskinhallinnan avulla projektiin liittyvää riskiä saadaan siis laskettua sen elinkaaren aikana. Jos riskinhallintaa ei käytetä, projektiin liittyvä riski kasvaa projektin edetessä. Tällainen tilanne on havainnollistettu kuvassa 2.2.



Kuva 2.2 Hallitsemattoman riskin vaikutus epävarmuuteen. Hallitsematon riski on korkeimmillaan projektin loppupisteessä. Mukailtu lähteestä [8, p. 97]

Hallitsemattoman riskin tapauksessa riski on suurimmillaan, kun kehitetty tuote julkaistaan. Tämä lisää myös epävarmuutta. Jos riski realisoituu, sillä on potentiaalisesti suuri vaikutus yrityksen toimintakykyyn. Muutoin kuvan 2.2 käyrät käyttäytyvät samalla tavalla kuin kuvassa 2.1.

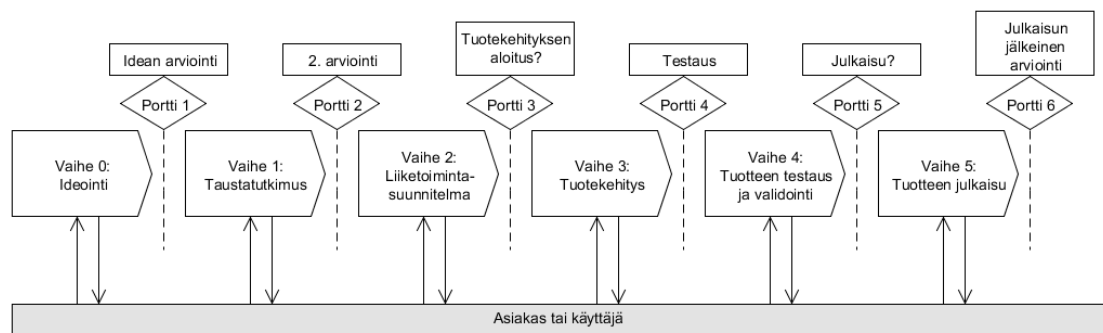
Innovoinnin asiantuntijat ovat jo vuosien ajan tiedneet, että suunnitteluprosessin toteuttaminen järjestelmällisesti on hyväksi lopputulokselle. Cooperin, Scott Edgettin ja Elko Kleinschmidin eri teollisuudenalojen yrityksille teettämän kyselytutkimuksen mukaan 75 %:lla vastanneista yrityksistä oli käytössään jonkinlainen innovointiprosessimalli. Kyselyn perusteella on pääteltävissä, että innovointiprosesseista on hyötyä, sillä yrityksen menestys korreloi selvästi käytössä olevaan innovoinnin organisoimisen tasoon. Yhteistä menestyneimmille yrityksille oli, että innovointiprosessi on selkeästi dokumentoitu. Tällöin tiedetään tarkasti, mikä seuraava kehitysvaihe tulee olemaan ja edistymistä pystytään helposti vertailemaan esimerkiksi edellisten projektien historiatietoihin. Menestyneissä yrityksissä innovointisuunnitelmien käyttö ei jäänyt ainoastaan suunnittelun tasolle, vaan niitä oikeasti myös hyödynnettiin – käyttöä valvottiin ja seurattiin. Edistymisen ja suunnitelman seuraaminen on johdon organisaatiokykyjen mittari. [9, p. 47–49]

Hyvä innovointiprosessimalli on skaalautuva ja sitä voidaan käyttää hyvin erilaisissa projekteissa. Mikään innovointiprosessi ei ole täysin samanlainen, joten suunnitelma ei saa olla liian jäykkä. Esimerkiksi täysin uuden tuotteen kehittäminen saattaa vaatia täysimittaisen prosessin läpikäymisen. Vanhan tuotteen jatkokehittämiselle saatetaan riittää prosessin läpikäyminen lyhemmän kaavan mukaisesti. Myös toimiva suunnit-

telman täytyy olla jatkuvassa kehityksessä. On pohdittava, mitkä käytetyistä menetelmistä ovat hyviä ja huonoja. Esimerkiksi byrokratian keventäminen on yleinen tavoite yritysten sisällä. Vain jatkuvalla kehityksellä voidaan saavuttaa merkittäviä kilpailuetuja markkinoilla. [9, p. 48–49]

2.2 Stage-Gate-innovointiprosessimalli

Eräs tuotteen innovointiprosessimalli on Cooperin suunnittelema Stage-Gate-malli. Sen avulla pyritään kuljettamaan idea valmiiksi tuotteeksi mahdollisimman tehokkaasti. Mallin tavoitteena on eliminoida epäonnistuneiden projektien tunnuspiirteet eli hätiköidyn suunnittelun, huonon johtamisen, huonon laadun ja pettävät aikataulut. [8, p. 83] Kuvassa 2.3 on esimerkki tyypillisestä Stage-Gate-kaaviosta.



Kuva 2.3 State-Gate-prosessin kulku kaavion avulla esitettynä. Prosessi koostuu erilaisista työvaiheista. Vaiheita erottaa arviointipisteet eli portit. Vaiheiden ja porttien selitykset käydään läpi luvussa 2.2.1. Kuva mukailtu lähteestä [8].

Kyseinen innovointiprosessimalli jakautuu vaiheisiin (stages) ja portteihin (gates). Vaiheiden tarkoituksena on kerätä tietoa projektista tai saavuttaa jokin muu tavoite, jotta voidaan edetä arviointipisteeseen eli porttiin. [8, p. 99] Jokaiselle portille saavuttaessa edellisessä vaiheessa täytyy olla tuotettuna sovitut asiat, esimerkiksi markkinatutkimus tai pohjatietojen kerääminen. Portilla vaihtoehtona on jatkaa seuraavaan tilaan, jos tarkasteltavan vaiheen tulokset vaikuttavat onnistuneilta tai lähettää projekti takaisin edelliseen vaiheeseen, jos tulokset eivät ole riittävän hyviä. Kolmas vaihtoehto porttien kohdalla on projektin hylkääminen tai siirtäminen myöhemmäksi. Päätösten tekemistä helpottaa, jos jokaiselle portille on ennalta määriteltäviä saavutettavia kriteereitä. [8, p. 101]

Portit ovat siis eräänlaisia luovutus- ja arviointipisteitä, joissa tarkastellaan projektin etenemistä. Kyse on koko projektin mittaisesta laadunvalvontaprosessista. [10, p. 50] Jokaiselle portille on määriteltävä ”portinvartija”, eli ryhmä tai ihminen, jotka arvioivat vaiheesta saadun tiedon ja tekevät etenemispäätöksen. Portinvartijat eivät ole staattisia, eli ne voivat vaihtua kesken projektin ja jokaisella portilla voi olla eri portinvartija. Esimerkiksi silloin jos projektiin liittyvä riski katsotaan kasvaneen ja tarvitaan lisää valtuuksia. Portinvartijoiden vaihtuminen on osa innovointiprosessin muuntautumiskykyisyyttä. On tärkeää, että portinvartijat myös osallistuvat projektin palaveriin ja että heille tuotetaan riittävästi informaatiota, jotta kyetään tekemään valistunut ja fak-

toihin perustuva etenemispäätös. Oleellista on, että yhteys asiakkaaseen on auki innovointiprosessin jokaisessa vaiheessa. Tarkoituksena on tehdä suunnittelusta läpinäkyvää ja helpottaa vaatimusten muuttumiseen reagointia. [10, p. 51–53]

2.2.1 Eri vaiheiden ja porttien merkitykset

Tässä luvussa käydään läpi kuvassa 2.3 esitetyn Stage-Gate-mallin eri vaiheiden ja porttien tarkoitukset. Ensimmäinen vaihe on idean keksiminen. Riippuen projektista tätä vaihetta ei vielä välttämättä lasketa viralliseksi vaiheeksi. Tämän vuoksi ensimmäistä vaihetta nimitetään usein vaihe 0:na. Kuitenkin yrityksessä idean keksiminenkin voi olla muodollinen prosessi, jolloin ideointi on itsenäinen vaihe. Ideointivaiheen tarkoituksena on saattaa projekti alkuun ja prosessi liikkeelle. Yhteen projektiin ei välttämättä riitä vain yksi idea, joten ideoita on hyvä olla kehitettynä useita valmiiksi. Hyvän tuotteen kehittämistä helpottaa, jos alkuperäinen idea on hyvä. [8, p. 103]

Vaihe 0:n jälkeen tulee ensimmäinen portti, joka on nimeltään idean tarkastelu. Portilla käydään läpi idean alustavaa potentiaalia, toteuttamiskelpoisuutta ja sopivuutta yritykselle. Tarkoituksena on tehdä päätös jatketaanko idean työstämistä. Tämä on ensimmäinen hetki, jolloin pohditaan resurssien sijoittamisen järkevyyttä. Portilla otetaan huomioon, miten idean tuote soveltuu yrityksen tuotevalikoimana, mitkä ovat markkina-tilanteen lähtötiedot ja miltä yrityksen tekninen osaaminen vaikuttaa suhteessa idean haastavuuteen. [8, p. 104]

Ensimmäinen varsinainen vaihe on nimeltään taustatutkimus. Tässä vaiheessa käydään huomattavasti syvällisemmin läpi, mikä on idean tekninen potentiaali ja markkinoilla vallitseva tilanne. Käyttökelpoinen tapa on luoda alustava markkinatutkimus ja käyttää hyväksi erilaisten kontaktien tarjoamaa informaatiota idean potentiaalista. Myöskään ei kannata jättää huomiotta kirjastojen ja Internetin tarjoamaa tietoa. Alustavan markkinatutkimuksen lisäksi tilassa on hyvä käydä läpi tarkemmin myös idean tekniestä toteuttamista. Alustava hinta-arvio sekä tuotteelle että innovointiprosessille on käyttökelpoinen tieto, jonka avulla on mahdollista tehdä päätöksiä ideoiden kohtaloista. Myös yrityksen sisäinen tekninen osaaminen ja tarpeet on syytä ottaa huomioon. Jos yrityksellä ei ole riittävästi osaamista idean toteuttamiseen, sen toteuttamisesta tulee todennäköisesti kallista ja epävarmaa. Jos tuote taas ei sovi yrityksen liiketoimintasuunnitelmaan, idean jatkotyöstäminen on tarpeetonta. [8, p. 105]

Portti 2 on nimeltään toinen arviointi. Käytännössä se on sisällöltään samanlainen kuin portti 1, mutta nyt tarjolla olevaa informaatiota on käsiteltävänä enemmän ja se on tarkemmin kerättyä. Jälleen kerran pohdittavina seikkoina on tekninen osaaminen, markkinoiden tilanne ja idean tuottopotentiaali. Tärkein kysymys on, onko idea kehityskelpoinen ja toteuttavissa? [8, p. 106]

Vaiheessa 2 on tarkoituksena luoda liiketoimintasuunnitelma. Taloudellisessa näkökulmassa tämä on erittäin tärkeä vaihe projektissa. Käytännössä kyseessä on jälleen hyvin samantyylinen vaihe kuin edellinen vaihe, mutta jälleen kerran informaatiota pyritään keräämään täsmällisemmin liiketoimintamahdollisuuksien näkökulmasta. Vaiheessa tehdään siis täsmällinen markkinatutkimus, jossa kartoitetaan tuotteen potentiaa-

li, kilpailijoiden asema markkinoilla, markkinoiden yleinen tila, tuotteen tarpeellisuus asiakkaiden näkökulmasta, idean tuottopotentiaali yritykselle, potentiaalisten asiakkaiden ja projektien kartoitus. Kuten vaiheiden etenemisestä huomataan, tarkoituksena on jatkuvasti syventää yrityksen omaa ymmärrystä projektista. [8, p. 107]

Portti 3 on nimeltään tuotekehitys, ja tarkoituksena on päättää markkinatutkimuksen tulosten perusteella onko järkevää siirtyä tuotekehitysvaiheeseen. Portti 3 on viimeinen portti, jossa projekti voidaan hylätä ennen taloudellisia riskejä. Kuten kuvista 2.1 ja 2.2 voitiin todeta, investoinnit alkavat kasvaa eksponentiaalisesti, mitä pidemmälle projekti etenee. [8, p. 109] Tämän vuoksi portti 3 on erittäin tärkeä vaihe yritykselle: huono päätös tarkoittaa potentiaalisesti suuria tappioita yritykselle. Liiketoimintasuunnitelman arvioinnin lisäksi on hyvä arvioida projektia myös kokonaisuutena. Jos tehty suunnitelma ja markkinatutkimus ovat järkeviä, idealla on potentiaalia ja myös edellisten vaiheiden tulokset ovat olleet positiivisia, voidaan projektissa siirtyä eteenpäin. [8, p. 109]

Vaihe 3 on nimeltään tuotekehitys. Tässä vaiheessa aloitetaan tuotteen tekninen työstäminen. Tuotekehitykseen kuuluu kehityssuunnitelman käynnistäminen, johon kuuluu esimerkiksi jaksotetut laboratorio- tai muut vastaavat testit. Tarkoituksena on varmistaa, että rakennettu tuote vastaa sille edellisissä vaiheissa asetettuja vaatimuksia. [8, p. 109] Jos tuote ei vastaa sille asetettuja vaatimuksia, se ei vastaa myöskään liiketoimintasuunnitelmaa. Tällöin tuotteen markkinapotentialin ei voida olettaa olevan sama kuin suunnitelmassa. Pääpaino tuotekehitysvaiheessa on teknisessä toteutuksessa ja tarkoituksena on saada vaiheen tuotteena osittain testattu prototyyppi. On kuitenkin tärkeää muistaa, että kosketus asiakkaaseen ja markkinoihin ei saa kokonaan katketa tuotekehitysprosessin aikana – yhteys on tärkeässä asemassa määrittäessä millaisia teknisiä painotuksia tuotteisiin tulee. Yhteyden säilyessä pystyy yritys myös helpommin reagoimaan muuttuneisiin toiveisiin tai vaatimuksiin. Tuotekehitysvaiheen jälkeen muutosten tekeminen voi olla hankalaa – osittain jopa mahdotonta ja vähintään kallista. [8, p. 109]

Koska tuotekehitysvaihe on hyvin tärkeä ja todennäköisesti työläin vaihe uuden tuotteen luomisessa, tämä kokonaisuus on usein järkevä jakaa pienempiin vaiheisiin. Vaiheet voivat olla samanlaisia tila- ja porttiyhdistelmiä kuin koko projektin mittakaavassa. Vaiheen sisäiset portit eivät kuitenkaan sisällä projektin hylkäys- tai jäädytyspäätöksiä, vaan niiden tarkoitus on jaksottaa ja tehostaa kehitysprosessia. Kun suunnittelu on selkeästi rakennettu, aikataulussa on helpompi pysyä. [8, p. 109]

Portti 4 on nimeltään testaus. Se on tuotekehitysprosessin osio, jossa projekti-ryhmä tutkii onko tuotekehitysvaiheesta valmistunut prototyyppi riittävän hyvä siirtymään testausvaiheeseen ja vikojen etsintään. Vaihtoehtona voi olla palaaminen takaisin tuotekehitykseen tai radikaalimmin projektin jäädyttäminen tai hylkääminen. Luonnollisesti projektin tässä vaiheessa näistä päätöksistä koituu jo merkittäviä kuluja yritykselle, sillä varsinkin tuotekehitykseen on jouduttu sitouttamaan merkittävästi resursseja. Joka tapauksessa portilla on jälleen kerran tutkittava edellisen vaiheen tulokset ja verrattava niitä ennako-oletuksiin. [8, p. 109]

Vaiheessa 4 prototyyppi kehitetään julkaisukuntoon, eli siitä pyritään löytämään kaikki mahdolliset viat ja puutteellisuudet. Vaiheen nimi on testaus ja validointi. [8, p. 110] Epäluotettavalla tuotteella on negatiivinen yrityksen maineeseen. Testivaiheessa löydetyn vian tai puutteellisuuden korjaaminen on halvempaa verrattuna projektin myöhempiin vaiheisiin. [11] Jotta mahdollisimman moni vika onnistuttaisiin paikantamaan, pyritään testausvaiheessa tekemään tuotteelle mahdollisimman monenlaisia erilaisia testejä. Onkin tärkeää, ettei testausvaiheessa keskitytä ainoastaan teknisiin testeihin, vaan otetaan mukaan myös käyttäjätestit ja käyttäjien mielipiteet – asiakkaiden mielipide on se, joka loppukädessä ratkaisee tuotteen onnistumisen. [8, p. 110]

Portti 5 eli julkaisu, on viimeinen kohta, jossa tuoteprojekti voidaan hylätä tai keskeyttää. Tämän portin aikana päätetään onko tuote julkaisuvalmis. Portin aikana analysoidaan edellisen tilan testien tulokset ja käyttäjien näkökulmat. [8, p. 111]

Vaihe 5 on tuoteprojektin viimeinen tila: tuotteen julkaisu. Tässä vaiheessa tuoteprojekti siirtyy tuotekehityksen parista markkinointiasiantuntijoille, jotka aloittavat tuotteen myymisen aikaisemmassa vaiheessa tehtyjen markkinointi- ja operaatiosuunnitelmien mukaisesti. Julkaisun jälkeenkään ei voida unohtaa tuotteen teknistä puolta ja tuotekehitystä – tuotetta on jälkiseurattava, eli mietittävä mahdollisia parannusehdotuksia, etsittävä virheitä ja pohdittava, synnyttääkö tuote uusia tarpeita eli mahdollisuuksia uusiin tuotteisiin. Tärkeää on myös pohtia, mikä projektissa on ollut onnistunutta ja missä vaiheessa tuotekehitysprosessia on eniten parannettavaa. Kriittinen pohdiskelu auttaa kehittymään ja tehostamaan omaa toimintaa markkinoilla. [8, p. 111]

State-Gate-malli pyrkii jäsentämään ja selkeyttämään tuotekehitysprosessia eli tekemään siitä helpommin hallittavan kokonaisuuden. Tuotekehitysprosessiin liittyy aina riskejä. Yritys sitouttaa projektiin resurssejaan – aikaa ja rahaa. Mitä pidemmällä tuotekehitysprosessi on, sitä enemmän siihen on sitoutuneena resursseja. Tuotekehitysprosessin noudattaminen on osa riskien hallintaa. Tuoteprosessit nähdään kaiken kaikkiaan tärkeänä osana nykyaikaisten yritysten johtamista [9, p. 8]. Pyrkimyksenä on lisätä tuotekehityksen tuottavuutta ja edesauttaa taloudellisten tavoitteiden saavuttamista. Uudet tuotteet auttavat yritystä säilyttämään kilpailukykyä ja luomaan etulyöntiasemaa kilpailijoihin nähden. Cooperin mukaan eräs yritysjohtaja on todennut osuvasti State-Gate-mallista, että se antaa yritykselle mahdollisuuden kokeilla uutta: joko onnistua tai epäonnistua, mutta samalla tarjoten joustavan mahdollisuuden siirtyä nopeasti uuteen projektiin ilman suuria tappioita. [9, p. 50]

3 TUOTERUNKOARKKITEHTUURIT

IEEE 1471-2000 on ohjelmistoarkkitehtuurien kuvaamista koskeva standardi, jonka mukaan ohjelmistoarkkitehtuuri on kohdejärjestelmän perusorganisaatio. Se sisältää kaikki järjestelmän osat ja niiden keskinäiset ja ympäristöön vaikuttavat suhteet, jotka ohjaavat järjestelmän suunnittelua ja kehitystä. [12] Eli arkkitehtuuri ei jaa ohjelmaa pelkästään ymmärrettäviin palasiin, vaan siihen liittyy myös osien väliset ja ulkopuoliset suhteet sekä koodin sisäiset rakenteet. Koskimies ja Mikkonen kuvaavat ohjelmistoarkkitehtuurin olevan ohjelman perustuslaki. [13, p. 18]

Arkkitehtuurin päätarkoituksena on pitää ohjelma selkeänä eli kontrolloida monimutkaisuutta. Ongelmat jaetaan pieniin loogisiin paloihin, joita on helpompi käsitellä kuin suuria kokonaisuuksia. Nämä arkkitehtuurin palaset keskustelevat keskenään ja muodostavat kokonaisuutena toimivan ohjelman. [14, p. 6] [15, p. 20] Erilaisia arkkitehtuurin osioita ovat esimerkiksi kommunikointitavat, rajapinnat, ja sisäiset prosessit [13, p. 19]. Arkkitehtuurien merkitys ohjelmistotuotannossa on kasvava, koska tuotettavat ohjelmat ovat yhä monimutkaisempia. Samalla kokonaisuuden hallinta on yhä haasteellisempaa. Nykyajan teollisuudessa ohjelmistotuotanto on usein rinnakkaista, joten insinööreiltä vaaditaan keskittymistä mahdollisesti useampaan eri projektiin. Onnistuneesti suunniteltu arkkitehtuuri on hyödyllinen ja uudelleenkäytettävä työkalu tulevaisuuden projekteille. [13, p. 15–16] Näin olleen on erityisen tärkeää, että arkkitehtuurin suunnittelu tukee projektille valittuja suunnittelulinjoja. [14, p. 312]

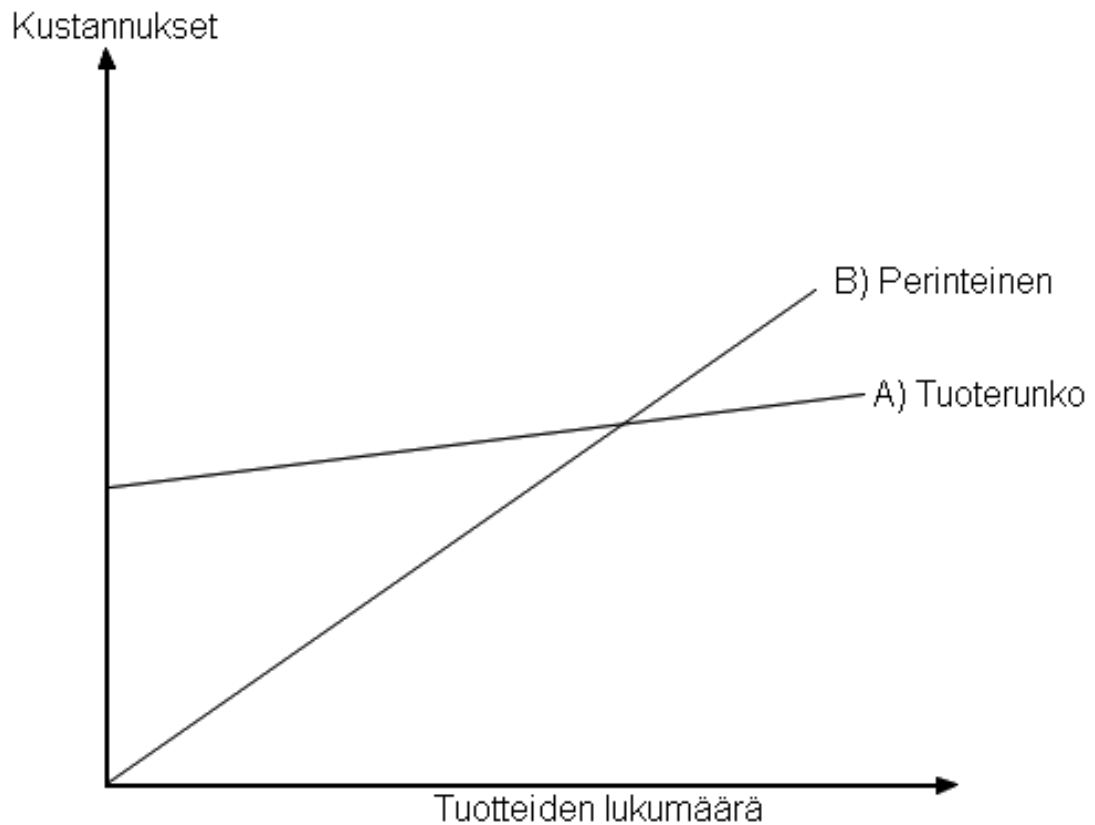
Ohjelmistoarkkitehtuurin täytyy tukea järjestelmän tavoitteita. Useimmiten arkkitehtuuri rakentuu järjestelmän tärkeimmän toimintaperiaatteen ympärille. Myös järjestelmän kriittisimmät rakenteet on otettava suunnittelussa tarkasti huomioon. Arkkitehtuureille asetetaan usein erilaisia käytännöllisiä vaatimuksia. Näitä ovat esimerkiksi muutettavuus, ylläpidon helppous, suorituskyky, testattavuus, uudelleenkäyttö ja luotettavuus, joka voi tarkoittaa esimerkiksi tulosten tarkkuutta, tietoturvaa, huollettavuutta tai häiriöiden sietokykyä. [15, p. 20] Vaikka suunnittelun lähtökohdat ovat aina toiminnallisia – mitä ohjelman on tarkoitus toteuttaa ja suorittaa – lopullisen arkkitehtuurin muodon sanelevat kuitenkin lopulta enemmän edellä esitetyt tavoitteet. [13, p. 21] Ohjelmiston tehtävänä on toimia asiakkaan työkaluna, mikä tarkoittaa sitä, että puutteellisia ominaisuuksia vaaditaan myöhemmin lisättäviksi. Asiakkailta on myös tapa kiertää huonosti tehtyjä ominaisuuksia, mikä voi johtaa tuotteen väärinkäyttöön. Kehityksessä on ymmärrettävä, ettei samanlainen lähestymistapa aina toimi, esimerkiksi eri maiden tai teollisuuden alojen kulttuurierot tuovat oman haasteensa suunnitteluun. [14, p. 9]

Uudelleenkäyttö on hyvin keskeinen asia ohjelmistotuotannossa. Jos ohjelmisto- ja pystytään käyttämään uudelleen, kehitystyössä saadaan säästettyä resursseja ja lyhennettyä kehitysaikaa. Uudelleenkäyttämisestä on tullut jopa välttämättömyys joillakin teollisuuden aloilla. [13, p. 157] Tuotannon hinnasta ja tehokkuudesta on tullut yhä tärkeämpiä vaikuttimia ohjelmistotuotantoon johtuen kiristyneestä taloudellisesta tilanteesta ja kilpailusta. Mitä nopeammin ja halvemmalla ohjelmisto saadaan kehitettyä sitä parempi. Uusiokäyttö muodostaa säästöjä erityisesti sen vuoksi, että ohjelmistotuotannon kustannukset painottuvat tuotekehitysvaiheeseen. Mitä enemmän tuotekehitysvaiheessa voidaan uusiokäyttää vanhoja komponentteja sitä vähemmän resursseja kuluu. Van Vliet vertaa ohjelmistotuotantoa perinteiseen rakennusteollisuuteen, jossa vastaavanlainen komponenttien uudelleenkäyttö ja kopiointi on mahdotonta. [14, p. 7–9] Tuoterunkoarkkitehtuuri on arkkitehtuuri, jonka avulla pystytään luomaan ohjelmistoja, jotka on suunniteltu uudelleenkäytettäväksi. [13, p. 157] Tuoterunkoarkkitehtuurin perustana on kuitenkin aina eri tuotteiden riittävä samankaltaisuus: jos tuotteet eivät ole riittävän samankaltaisia, on riittävän kattavan tuoterunkoarkkitehtuurin suunnittelemisen haastavaa. [16, p. 261] Suunnitelmallinen uudelleenkäyttäminen pohjautuu aina arkkitehtuuritason ratkaisuihin. [13, p. 157]

Tuoterunkoarkkitehtuuriratkaisulla pyritään siis toteutetun ohjelmistoarkkitehtuurin suunnitelmalliseen uudelleenkäyttämiseen samankaltaisissa tuotteissa. Esimerkki samankaltaisesta tuotteesta on ohjelmistosta tehdyt erilaiset versiot eri asiakkaille tai ympäristöille. [16, p. 261] Tuoterunkoarkkitehtuurille voidaan nähdä useita etuja jo mainitun taloudellisen näkökulman lisäksi. Ensimmäiseksi tuoterunkoarkkitehtuurin avulla saadaan nopeutettua tuotteiden valmistamista. Luonnollisesti syy tähän on, että osa tarvittavasta koodista on valmiina ja tuotteen arkkitehtuuri – minkä rakentaminen on projektin haastavin ja aikaa vievin osuus – on valmiiksi suunniteltu. Metodilla on myös positiivisia vaikutuksia koodin laatuun: koska koodia on osaksi käytetty jo aiemmissa tuotteissa, on se osittain myös testattu. Toisin sanoen tuoterunkoarkkitehtuuri helpottaa erilaisten työkalujen siirtämistä projektista toiseen, sillä ympäristöt ja työkalut ovat samankaltaisia. [13, p. 161] Lisäksi samankaltaiset ohjelmistot yksinkertaistavat ohjelmistojen ylläpitoa, sillä myös ylläpitotoiminnot samaan arkkitehtuurin perustuvissa tuotteissa ovat samankaltaisia. [16, p. 261] Edut eivät kuitenkaan jää ainoastaan ohjelmistopuolelle, vaan tuoterunkoarkkitehtuurikehityksellä on etuja myös organisaatiotasolla. Koska tuoteprojektit ovat rungoltaan samanlaisia, niiden hallinnointi on helpompaa ja toistuvia prosessimalleja voidaan soveltaa useissa projekteissa. Koskimies ja Mikkonen kuvailevat osuvasti tuoterunkoarkkitehtuuripohjaisen kehitystavan standardoivan sekä tuotteita että niiden kehitysprosesseja. Kyseinen asia voidaan ymmärtää myös ohjelmistotuotannon automaationa. [13, p. 161]

Organisaation kannalta ajateltuna tuoterunkoarkkitehtuurin tavoitteena ovat siis taloudelliset ja tuotannolliset parannukset: yhteen tuotteeseen käytettävän työmäärän vähentäminen ja siten nopeampi pääseminen markkinoille – asia, joka State-Gate-mallia käsiteltäessä todettiin hyvin tärkeäksi. Kehitys- ja valmistusprosessin standardoitumisella on positiivisia vaikutuksia tuotteiden laatuun. [17, p. 241]

Oleellinen kysymys tuoterunkoarkkitehtuuria kehittäessä on, milloin tällainen arkkitehtuuri kannattaa kehittää? Kuten jo aiemmin todettiin, pääasia on kehitettävien tuotteiden samankaltaisuus, jotta arkkitehtuuri saadaan ylipäättään kehitettyä. [16, p. 261] Kuvassa 3.1 on esitetty miten ohjelmistotuotteiden kehityskustannukset vaihtelevat perinteisen ja tuoterunkoarkkitehtuuriin pohjautuvan ratkaisun välillä.



Kuva 3.1 Tuoterunkoon perustuvat tuotannon aloituskustannukset ovat suuremmat verrattuna perinteiseen tuotantoon. Mitä enemmän tuotteita tehdään sitä kannattavampaa tuoterunkoon perustuvatuotanto kuitenkin on. Mukailtu lähteestä [18]

Tuoterunkoarkkitehtuurin kehittämisellä on suuremmat alkukustannukset kuin perinteisellä ohjelmistotuotannolla. Kummankin mallin kustannukset nousevat lineaarisesti suhteessa tuotteiden lukumäärään, mutta perinteisellä tavalla nousun kulmakerroin on suurempi. Kuvaajat leikkaavat toisensa ja tuoterunkoarkkitehtuuriin perustuvasta tavasta tulee pitkällä aikavälillä halvempi vaihtoehto. Kuva visualisoi väitteen siitä, että tuoterunkoarkkitehtuuripohjainen kehitystapa vaatii enemmän tuotteita, jotta se muuttuu kannattavaksi. Kehitystapa vaatii siis pitkäjänteistä ajattelua: hyödyt eivät ole välittömästi havaittavissa. [13, p. 161]

Tuoterunkopohjaisessa kehityksessä onkin ajattelutavan siirryttävä yksittäisen tuotteen kehittämisestä tuoteperhekohtaisen kehityksen ajattelumalliin. [17, p. 243] Tuoteperhe on toiminnaltaan ja rakenteeltaan samankaltaisten ja tietylle sovellusalueelle toteutettujen sovellusten tai tuotteiden joukko. [13, p. 160] [16, p. 261] Tuoteperheissä samankaltaisten elementtien uudelleenkäyttäminen on harkittua ja suunnittelua – usein tuotteilla on myös joitain täysin samanlaisia elementtejä, mutta elementit ovat usein myös erilaisia tuoteperheen sisällä. Asia on otettava huomioon jo tuoteperheen suunnit-

teluvaiheessa, jotta tuotteiden toteuttaminen olisi mahdollisimman vaivatonta. [14, p. 70–71] Kun tuoteperheen periaatteita verrataan tuoterunkoarkkitehtuurin vaatimiin periaatteisiin, huomataan niiden olevan hyvin samankaltaiset. Tämän vuoksi tuoteperheille tuoterunkoarkkitehtuurin suunnitteleminen on järkevää.

3.1 Tuoterunkoarkkitehtuurin tuotantoprosessi

Tuoterunkoarkkitehtuurin toteuttaminen ja suunnittelu ei ole ainoastaan tekninen ponnistus, vaan siihen liittyy myös taustaorganisaation toiminta: yleinen organisointi ja käytetyt suunnittelutoimintamallit. Ilman järjestelmällistä toimintatapaa tuoterungon etuja ei saada tehokkaasti esille. Tuoterunkoarkkitehtuurin toteutus voidaan jakaa seuraaviin vaiheisiin: esitutkimusvaihe, alustankehitysprosessi ja tuotekehitysprosessi. [13, p. 164]

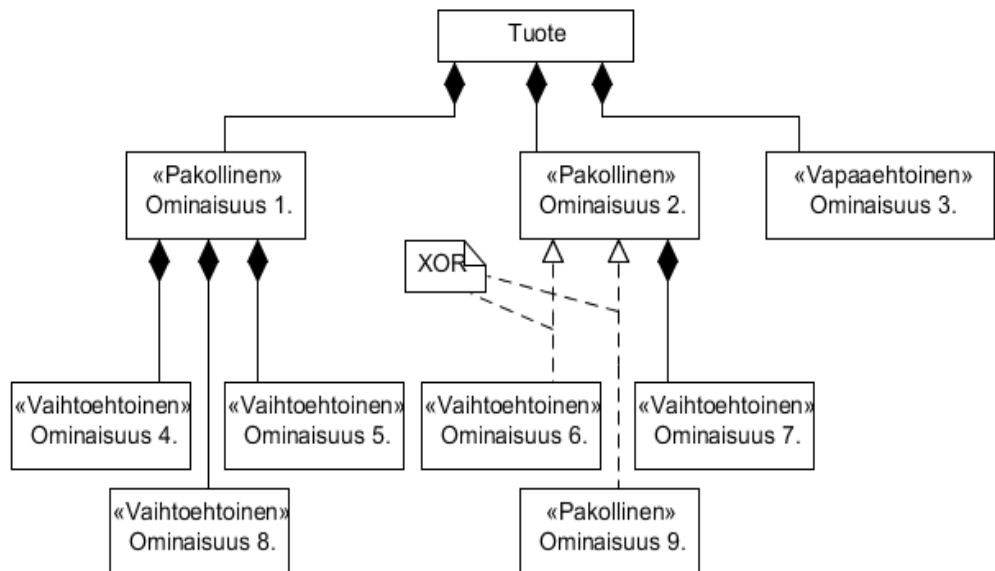
Kuten toisen luvun State-Gate-mallissa, tuoterunkoarkkitehtuurin kehitysprosessissa kaikki kehitysvaiheet ovat yhtä arvokkaita. Jos jokin vaiheista epäonnistuu täysin, lopputuotteessa on todennäköisesti jotain vialla. Esitutkimusvaihe on hyvin samankaltainen kuin State-Gate-mallin ensimmäiset tilat. Pääpaino on tarvittavien toiminnallisuuksien selvittämisessä. Työkaluna voidaan käyttää muun muassa markkinatutkimuksia tai haastatteluja. [19, p. 211–212] Esitutkimusvaiheessa onkin pohdittava tarkasti, onko runko tarpeellinen ja hyödyllinen tulevaisuudessa. [13, p. 164] Jos runko ei ole kannattava esimerkiksi vähäisen tuotantovolyymin vuoksi, tuoterunkoa ei kannata toteuttaa. Lisäksi organisaatiotasolla on pohdittava kuinka kauan kehitykseen kuluu aikaa, miten kehitysprojektin dokumentointi toteutetaan, kuinka ohjelmointityö organisoidaan, millaisia kustannuksia toteutus teettää ja millaiset laatuvaatimukset projektille asetetaan. [14, p. 42] Ohjelmointi ei ole muusta projektista erillinen osuus, sen täytyy tukea standardeja, ottaa huomioon turvallisuusvaatimukset ja mahdolliset yrityksen sisäiset asetukset. [14, p. 35] On ymmärrettävä millaisen ohjelmiston ongelman ratkaisu vaati ja käytettävä vakiintuneita ohjelmointitapoja, mikä mahdollistaa laadukkaan koodin. [16, p. 264] Esitutkimusvaiheen tärkein anti ohjelmiston kannalta on kuitenkin ominaisuuksien ja vaatimusten kartoittaminen. [20, p. 60] Ominaisuus on looginen toimintakokonaisuus, jonka toiminta on määritelty toiminta- ja laatuvaatimuksilla. Harva ominaisuus on täysin itsenäinen, vaan ominaisuuksilla on keskinäisiä riippuvuussuhteita. [19, p. 194]

Roope Parviaisen diplomityössä *Tuoterunkoon perustuvan tietovarastoratkaisun kehittäminen* (TTY, 2015), käydään läpi erään yrityksen tuoterungon kehitysprosessin vaiheet. Vaatimusten tunnistaminen voi tapahtua esimerkiksi käsitekartalla: ominaisuuksia aletaan pohtia painottuen tuotteen keskeisimpiin vaatimuksiin. Erilaisia vaatimuksia voidaan löytää suoraan asiakkailta, yrityksen aiemmista projekteista, liiketoimintatuntemuksen ja muiden kokemusten kautta. Olennaista on pyrkiä ottamaan vaatimuksilla huomioon mahdollisimman laaja asiakaskunta ja tulevaisuuden tarpeet. [20, p. 60–61] Osa vaatimuksista on pysyviä ja osa tuotekohtaisesti muuttuvia ominaisuuksia. [13, p. 166] Liiallinen asiakkaiden tarpeisiin keskittyminen voi johtaa harhaan omis-

ta resursseista ja osaamisesta. Parviaisen haastatteleman asiantuntijan mukaan tuoterunkoarkkitehtuuria ei pidä määritellä liian tarkasti yhden asiakkaan kanssa, vaan muodostettu kokonaisratkaisu tarjotaan asiakkaalta toiselle sellaisenaan. Näin pyritään välttämään liian erikoistunut arkkitehtuuri. [20, p. 60–61]

Vaatimusten määrittelyyn liittyy siis paljon kompromisseja ja taustaselvitystä. Liian suppea ratkaisu tuottaa lisätyötä tulevaisuudessa ja laaja ratkaisu voi olla omille resursseille liian hankala. Toisaalta jos asiakkaan tarpeita ei oteta riittävän laajasti huomioon, syntyy runkoon muutostarpeita, jotka teettävät aina lisätöitä. [20, p. 60–61] On osattava rajata tuoterunkoarkkitehtuurin toimialue ominaisuuksien avulla. Myös tämä on kompromissien tekemistä: liian tiukka rajausta rajoittaa alustan kasvupotentiaalia ja laajennettavuutta. Liian löyhä rajausta taas aiheuttaa toteutuksen tehottomuutta. [20, p. 75]

Kun vaatimukset on saatu määriteltä, on syytä pohtia mitkä vaatimuksista ovat toistuvia ja mitkä eivät. Eli mitkä vaatimuksista ovat mahdollisesti tuoterunkoarkkitehtuuriin kiinnitettäviä ja mitkä niistä on syytä jättää tuotekohtaisiksi. Jotkin vaatimukset ovat pysyviä tuotteesta toiseen, toiset voivat toteutua osittain ja jotkin voivat vaihdella kokonaan tuotteesta toiseen. Näiden eroavaisuuksien jakaantumiseen voidaan käyttää piirremallia. Kuvassa 3.2 on esitetty esimerkki piirremallista. [13, p. 170]



Kuva 3.2 Piirremallissa esitetään tuotteen ominaisuuksien jakautuminen pakollisiin, vapaaehtoi-
siin ja vaihtoehtoihin ominaisuuksiin. Ominaisuudet voivat jakautua aliominaisuuksiin. Kuva mu-
kailtu lähteestä [13, p. 170].

Piirremallin ideana on siis käytännössä kuvata arkkitehtuurin eri osioita ja niiden valinnaisuutta. Piirremallissa on esitettynä tuotteeseen tulevat pakolliset ominaisuudet, vaihtoehtoiset ominaisuudet ja vapaaehtoiset ominaisuudet. Mallissa on myös esitetty miten tuotteen eri ominaisuudet jakaantuvat myös itsessään erilaisiin ominaisuuksiin. Riittävän laaja kuvaus teknisistä ominaisuuksista, riippuvuuksista ja jatkuva muutosten seuraaminen, helpottaa tuoterunkoarkkitehtuurin tekemistä ja myöhemmin myös ylläpi-
toa. [20, p. 75]

Muunneltavuus, laajennettavuus, räätälöitävyys ja uudelleenkäyttö ovat tuoterunkoarkkitehtuurien tärkeimpiä ominaisuuksia. Ne voidaan saavuttaa erilaisilla lähestymistavoilla. Lähtökohta on kuitenkin hyvä suunnittelu ja asioiden huomioonottaminen mahdollisimman aikaisessa vaiheessa. Esimerkiksi voidaan käyttää arkkitehtuuripohjaista ratkaisua, kuten kerrosrakenteinen arkkitehtuuri, jossa ohjelman eri kerroksilla on erilainen toimintatarkoitus. Toinen ratkaisumalli on plugin- tai modulaarirakenteinen ohjelma. Toisin sanoen tuoterunko jaetaan erilaisiin komponentteihin, joita voidaan siirrellä ohjelmasta toiseen ja ohjelmaan luodaan rakenteita ja rajapintoja, joihin näitä modulaarisia komponentteja voidaan vaivatta liittää. Näin varioitavien kohtien määrää saadaan helpommin hallittua ja rajattua. [20, p. 67] Ideaalisesti ajateltuna komponentit ovat arkkitehtuurille niin sanottuja black-box-komponentteja, eli komponentit tarjoavat oman rajapinnan, johon runkoarkkitehtuuri pystytään liittämään eikä komponenttien sisältämästä koodista tarvitse olla tietoa. [19, p. 224] Komponentteja luodessa on hyvä muistaa, että yleensä pienempiä komponentteja on helpompi käyttää uudestaan kuin suuria, koska niissä on vähemmän riippuvaisuuksia kuin suurissa. Toisaalta suurten komponenttien luominen ja uudelleenkäyttäminen säästää potentiaalisesti paljon resursseja. Tärkeää on löytää sopiva tasapaino komponenttien laajuudessa. [19, p. 219] Edelleen on kuitenkin syytä painottaa, ettei pelkästään teknisten seikkojen suunnittelu riitä hyvän arkkitehtuurin toteuttamiseen. Lisäksi tarvitaan laajaa sovellusalueen tuntemusta. Uudelleenkäytettävyys on aina sisäänrakennettu ominaisuus. [20, p. 67]

Esitutkimuksen ja vaatimusten määrittelyn jälkeen seuraava vaihe on alustakehitysprosessi, jonka tarkoituksena on toteuttaa sovellus annettujen vaatimusten mukaisesti. [13, p. 164] Tuoterunгон rakentamiselle on erilaisia vaihtoehtoja. On mahdollista ottaa esimerkiksi olemassa olevan toteutuksen komponentit tuoteperheen perustaksi ja tehdä näiden kehiksen varaan oma tuoteperhe. Toinen vaihtoehto on rakentaa tuoterunko pikkuhiljaa. Eli rungon seuraava versio tarjoaa enemmän toimintoja kuin aikaisempi versio. Kolmas vaihtoehto on rakentaa tuoterunko tyhjästä – joko ensin rakentamalla tuoterunko ja sen jälkeen ohjelma tai toisin päin rakentamalla ensin tuote ja sen jälkeen runko. [13, p. 163] Joka tapauksessa tavoitteena on konkretisoida arkkitehtuuri.

Kehitysprosessiin liittyy oleellisesti ohjelmoinnin laatu. Laadun lisääminen jälkikäteen on haastavaa, joten ohjelmoinnin laatuvaatimusten huomioiminen pitää olla mukana prosessin alusta asti. Laatu on pitkällä tähtäimellä avain menestykseen markkinoilla ja huonolaatuiset tuotteet tuovat helposti huonon maineen yritykselle. Koska ohjelmistot ovat kehityksessä jatkuvasti monimutkaisemmiksi ja suuremmiksi, laadun huomioiminen on myös tästä syystä yhä tärkeämmässä asemassa. Koodin laatua on kuitenkin hankala mitata, koska sille ei ole absoluuttisia mittareita – jokaisella ihmisellä voi olla erilainen kuva laadukkaasta koodista. Testaajan mielestä ohjelma on laadukas, kun se täyttää ennalta määritellyt vaatimukset. Käyttäjän mielestä laatu taas voi olla esimerkiksi se miten tuote toimii lopullisesti käyttötarkoituksessaan. Ohjelmoijan mielestä laatu voi olla kommentoitua ja dokumentoitua koodia. Laatua ei tuotteeseen saa ilmaiseksi, vaan sitä on aktiivisesti tuotettava analysoinnin, testauksen, prosessimallien ja suunnittelun avustuksella. [14, p. 109] Esimerkiksi alustankehitysprosessissa testaa-

mista on suoritettava mahdollisimman monelta eri näkökannalta. On kiinnitettävä huomiota erilaisiin ympäristöihin, joissa lopputuotetta tullaan käyttämään ja sitä kautta pyrkiä löytämään mahdollisia puutteellisuuksia ja vikoja. [20, p. 62] Tuoterunkoarkkitehtuurien muunneltavuus tuo vaikeuksia testauksen toteuttamiseen. Valmiita tuotteita joudutaan usein testaamaan itsenäisinä kokonaisuuksina, koska tuoterunkoarkkitehtuuri ei vastaa kaikkiin ominaisuuksiin. Käytännössä testaaminen on hyvin työläs prosessi. [13, p. 172]

Alustankehitysprosessin tuloksena on toimiva ja testattu tuoterunkoarkkitehtuuri. Viimeisenä suorituvaiheena tuoterunkoarkkitehtuurin rakentamisessa on tuotekehitysprosessi, jonka tarkoituksena ei ole enää kehittää runkoa vaan siihen liittyviä erilaisia tuotteita. Eli tässä vaiheessa arkkitehtuuri siirtyy tuotantovaiheeseen, tuottamaan arkkitehtuurin suunniteltuja käytännön sovelluksia. [13, p. 164] Yksittäiset sovellukset luodaan asiakkaan tarpeiden mukaisesti, ja ne noudattavat aiemmissa vaiheissa määriteltyjä vaatimuksia. Yksittäisten tuotteiden kehittäminen kertoo toimiiko tuoterunkoarkkitehtuuri oikein valmiuksiensa edellyttämällä tavalla ja johdonmukaisesti. On tärkeää, että koko prosessin ajan tietoisuutta arkkitehtuurista jaetaan koko organisaation kesken, jotta vältetään osaamisen keskittyminen pienelle osalle yritystä. Käytännössä projektit tehdään aina yhteistyössä erilaisten asiantuntijoiden kanssa. [20, p. 62]

3.2 Tuoterunkoarkkitehtuurien jatkuva kehittäminen

Ohjelmistot eivät ole muusta maailmasta irrallisia kokonaisuuksia, vaan myös niiden on pystyttävä mukautumaan muutoksiin pysyäkseen käyttökelpoisina. Evoluution mukana tulee kuitenkin aina ongelmia, sillä kun suunniteltua arkkitehtuuria muutetaan, sen rakenne rikkoutuu ja käytännössä ohjelma muuttuu monimutkaisemmaksi. Monimutkaisuus taas lisää virhealttiutta. Tämä on ohjelmistojen evoluution perusongelma, joka korostuu mitä suurempi järjestelmä on kyseessä. [16, p. 66–67]

Ohjelmien muutoksia ei voi kuitenkaan välttää, koska ympäristöstä tulee muuospaineita ohjelmistoille. Paineita teettävät esimerkiksi lakimuutokset ja standardit, jotka voivat vaikuttaa esimerkiksi turvamääräyksiin ja rajapintoihin. [16, p. 66–67] Toinen luonnollinen suunta, josta muuospaineita tulee, ovat asiakkaat: he saattavat vaatia esimerkiksi uusia toiminnallisuuksia. Kilpailullinen markkinatilanne saattaa vaatia asiakkaiden toiveiden täyttämistä. On myös ymmärrettävä, että tehdyt muutokset luovat usein itsessään uusia muutostarpeita johtuen monimutkaisuuden lisääntymisestä. [16, p. 271] Yleisin muutostarve tuoterunkoarkkitehtuureissa on uuden ominaisuuden tai piirteen lisääminen. Jos ominaisuus sopii hyvin suunniteltuun arkkitehtuuriin ja arkkitehtuurin rajapinnat tukevat ominaisuutta tai muutosta, sen lisääminen on yksinkertainen prosessi. Ratkaisuna on lisätä tuoterunkoarkkitehtuurin liittyvään tuoteperheeseen uusi tuote. Uusien tuotteiden liittäminen tuoterunkoon on välttämätöntä kehityksen takaamiseksi [13, p. 163]. Jos uuden tuotteen lisääminen tuoteperheeseen ei onnistu, on ongelma ratkaistava eri tavalla. [16, p. 271–273] Mahdollinen ratkaisu ongelmaan on arkkitehtuurin sisäisen komponentin ominaisuuksien kehittäminen vastaamaan uusia tarpeita.

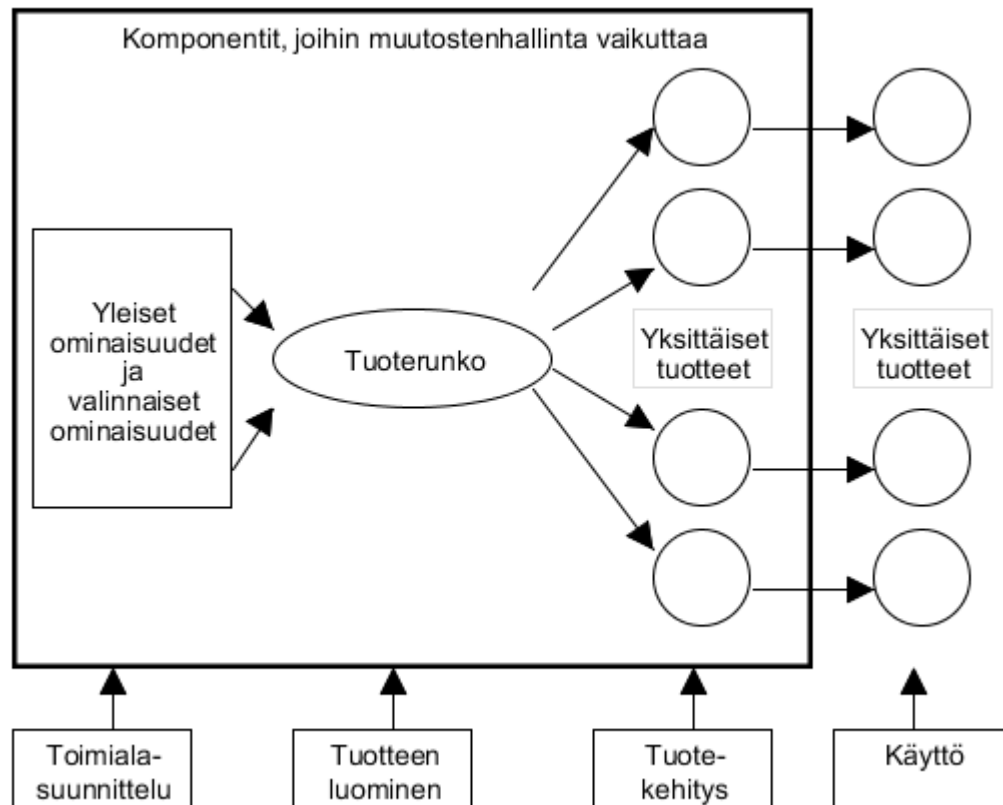
Tämä on pienin mahdollinen muutos, joka arkkitehtuuriin voidaan tehdä. Tällaisen muutostarpeen voi luoda esimerkiksi käyttöjärjestelmän päivittyminen. Käyttöjärjestelmään voi esimerkiksi tulla ominaisuuksia, jotka ovat jo sisällytettynä muualle arkkitehtuuriin. Tällöin näiden ominaisuuksien poistaminen arkkitehtuurista on perusteltua, koska samoja ominaisuuksia ei tarvita kahdessa eri systeemissä. [16, p. 271–273]

Vaatimuksena voi olla myös uuden tuotteen lisäämisen tuoteperheeseen. Jos tuoterunko ei valmiiksi tue uutta tuotetta, yksi mahdollinen ratkaisu on muuttaa arkkitehtuuria siten, että se tukee uutta tuotetta. Tämä päätös tarkoittaa joko vanhan tuoterungon päivittämistä tai eriyttämistä rinnakkaiseksi versioksi. Eriyttäminen voi olla tarpeellista esimerkiksi silloin, jos tuotteen tarvitsema ominaisuus haittaa jotakin arkkitehtuurin ominaisuutta. Toinen syy erittymiseen voi olla tärkeä asiakas, jonka vaatimukset eriävät muiden asiakkaiden vaatimuksista. Omalla arkkitehtuurilla tärkeimmän asiakkaan toiveet on helpompi toteuttaa. [16, p. 271–273]

Viimeinen ratkaisu uuden ominaisuuden lisäämiselle on rakentaa kokonaan uusi tuoterunkoarkkitehtuuri, joka perustuu vanhaan arkkitehtuuriin. Päätös voi tulla ajan-kohtaiseksi silloin kun tuoteperheeseen liittyvien tuotteiden vaihtelevuus kasvaa liian suureksi, eikä arkkitehtuuri pysty enää tehokkaasti suoriutumaan tästä vaihtelusta. Ongelma syntyy helposti ajan kanssa, sillä tuoteperheeseen liittyvien tuotteiden lukumäärä usein kasvaa ajan kuluessa. Toinen yleinen syy on yritysten toimintatapojen muutokset, esimerkiksi yrityskauppojen vuoksi tai markkinamuutokset. Myös yrityksen koko on usein ratkaisevassa asemassa. Johtuen ihmisten kulttuurien, tapojen ja tarpeiden eroista voi hyvinkin olla järkevämpää tehdä eri maantieteellisille alueille omat tuoterunkoarkkitehtuurit. [16, p. 271–273]

Edellisistä kappaleista voidaan päätellä, että tuoterunkoarkkitehtuurien evoluution ja kehityksen suurin ongelma on hallita arkkitehtuuriin tulevia muutoksia. Runkoarkkitehtuuri liittää tuoteperheen yhteen ja tämän vuoksi evoluutio on rajoitetumpaa kuin yksittäisen ohjelman kehityksessä – muutosten pitää toimia useassa eri ohjelmassa. Mitä paremmin muutostarpeisiin on varauduttu kehityksessä, sitä helpompi niitä on hallita. [16, p. 271] [13, p. 172] Varautuminen on työlästä ja hankalaa, minkä vuoksi tuoterungot usein tukevat aluksi vain pieniä tuoteperheitä. Osaamisen kehittyessä ja rungon laadun parantuessa runkoon on mahdollista liittää enemmän ohjelmia. [13, pp. 163, 168]

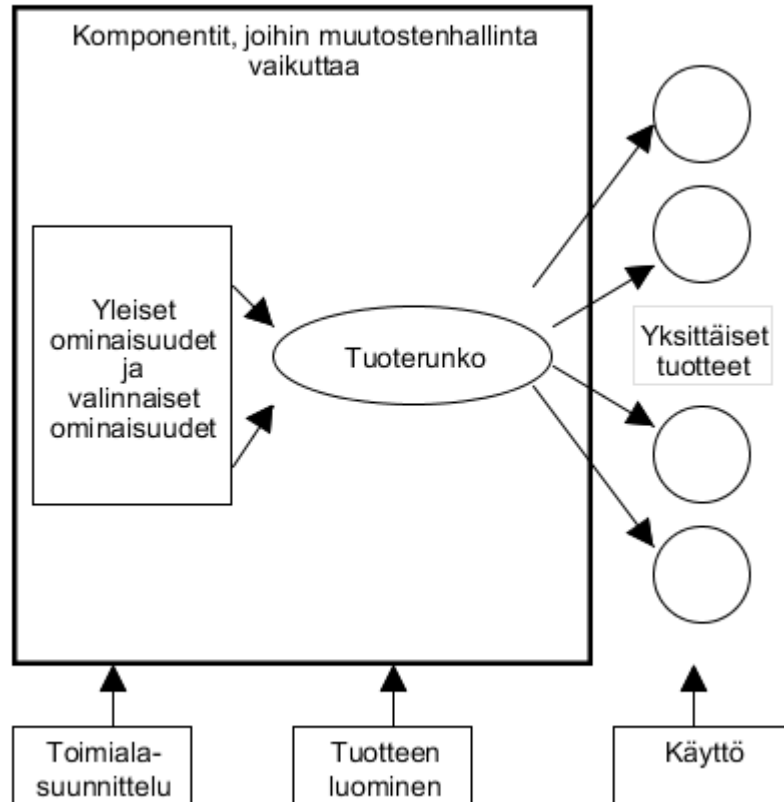
Charles Krueger nimittää tuoterunkoarkkitehtuurin muutosten hallintaa multidimensionaaliseksi hallintaongelmaksi. Perinteisessä tuotesuunnittelussa muunneltavuutta hallitaan vain suhteessa aikaan. Tuoterunkoarkkitehtuurien parissa hallintaongelma on ajan suhteen lisäksi myös variaatiot eri tuotteiden välillä. [21, p. 1–2] Myös muutoksia niin arkkitehtuurin tiedostoissa, komponenteissa ja runkoon liittyvissä tuotteissa täytyy hallita. [21, p. 3–4] Yleinen tapa vastata muutostenhallintahaasteeseen on esitetty kuvassa 3.3.



Kuva 3.3 Yleinen tapa hallita tuoterunkoarkkitehtuurin muutostenhallintaa. Oleellista on, että muutostenhallinta kohdistuu myös yksittäisiin tuotteisiin, jotka jatkavat omaa elinkaartaan. Ei pelkästään tuoterunkoon. Mukailtu lähteestä [21, p. 5].

Kuvassa laatikon sisällä olevat vaiheet kuuluvat muutostenhallintaprosessiin. Ensimmäinen vaihe on yleisten ja variointikomponenttien suunnittelu. Tämä osuus kuuluu toimialakohtaiseen suunnitteluun. Siihen tulisi sisältyä arkkitehtuurin yleiset asiat, yleiset variaatiot, tuoteperheen arkkitehtuuri, mutta ei yksittäiset tuotteet. Toinen vaihe on tuotteen luominen, jonka jälkeen siirrytään yksittäisen tuotteen tuotekehitykseen. Tässä mallissa tuotteet ovat ikään kuin yksilöllisiä ulostuloja, jotka elävät omaa elämäänsä, eivätkä kuulu yleiseen muutostenhallintaan. Tämän kehitystavan ehdoton heikkous on se, että tuotteeseen tehtävä muutos ei millään tavalla heijastu takaisin tuotantoprosessiin tai muihin tuotteisiin. Samoin, jos tuotantoprosessiin tehdään jälkikäteen muutoksi, eivät nämä muutokset välttämättä ole yhteensopivia valmiiden tuotteiden kanssa. [21, p. 5–6]

Kruegerin ehdottama parannus tähän variaation hallintaan esitetään kuvassa 3.4.



Kuva 3.4 Toinen tapa tuoterungon muutostenhallintaan. Tuotteet otetaan suoraan tuoterungosta suoraan käyttöön. Yksittäisten tuotteiden muutoksia ei hallinnoida yleisesti vaan asiakaskohtaisesti. Kuva mukailtu lähteestä [21, p. 6].

Tässä mallissa muutosten hallinta on rakennettu tiukemmin tuoteperheen ympärille. Toisin sanoen tuotteet otetaan käyttöön suoraan tuotekehityksen jälkeen ilman ylimääräisiä modifikaatioita. Edelliseen prosessiin verrattuna tässä kehitystavassa on yksi vaihe vähemmän. Etuna on se, että mallissa varioidaan ainoastaan yhtä tuotetta – muutokset tehdään suoraan tuoteperheen juureen. Näin vältetään tilanne, jossa pitää huolehtia useasta eri tuotteesta. Pääidea on varioida tuotantoa, ei yksittäistä tuotetta. [21, p. 7]

Muunneltavuuden hallinta tuoterunkoarkkitehtuurissa on sen suurin heikkous. Ongelman ratkaisemiseksi on tärkeää käyttää hyväksi koko organisaatiota ja uskoa pitkäjänteiseen työhön. Asia johon nykyaikana ei yritysten kärsivällisyys aina riitä, koska välittömät tulot ovat tärkeässä asemassa. [13, p. 174]

3.3 Tuoterunkoarkkitehtuurin ylläpito

Ohjelmiston muunneltavuuteen ja kehitykseen liittyy vahvasti ylläpito. Usein muutostarpeet ovat ylläpidollisia toimintoja tai ylläpidolliset toiminnot voivat laukaista muutostarpeen. Ohjelmien kehityksestä jopa 80 % on ylläpitoa ja tuotekehitystä noin 20 %. [19, p. 282] Ohjelmistoissa on aina virheitä – moni virheistä on niin sanottuja piilovirheitä, joita ei löydä testaamalla helposti. Virheet on korjattava myös julkaisun jälkeen. Lisäksi mikään ohjelma ei muutospaineiden vuoksi toimi sellaisenaan ikuisesti, vaan

vaatii työtä. Ylläpito ei ole ainoastaan virheiden korjaamista, vaan siihen liittyy myös muita toimintoja. Ylläpito voidaan neljään eri tehtävään, jotka ovat:

- korjaava ylläpito
- mukauttava ylläpito
- täydellistävä ylläpito
- ennaltaehkäisevä ylläpito.

Korjaavan ylläpidon tehtävänä on korjata toimitetussa ohjelmassa esiintyvät virheet. Mukauttavan ylläpidon tehtävänä taas on muuttaa ohjelma vastaamaan muuttuneita vaatimuksia kuten uutta käyttöjärjestelmää. Täydellistävän ylläpidon tehtävänä on lisätä erilaisia toimintoja tai esimerkiksi parantaa suorituskykyä. Ennaltaehkäisevän ylläpidon tarkoituksena on parantaa järjestelmän ylläpidettävyyttä. Tämä voi tarkoittaa esimerkiksi koodin dokumentoinnin parantamista. [14, p. 16–17] Suurin osa ylläpidosta on täydellistävää ylläpitoa [14, p. 461]. Tämä on hyvä osoitus siitä, kuinka ohjelmistojen kehitys on käytännössä syklinen prosessi, jossa palataan korjaamaan ja parantamaan vanhaa.

Ylläpitoon liittyy kuitenkin useita ongelmia. Yleisin syy ylläpidon vaikeuteen on huonosti tehty koodi, mikä voi johtua esimerkiksi kokemattomasta ohjelmoijasta. Huono koodi voi näkyä esimerkiksi sekavana rakenteena. Toinen yleinen syy ongelmiin on, ettei ohjelmoijalla ole ollut riittävästi tietoa kohdejärjestelmästä. Tällöin ohjelman toiminnot eivät välttämättä vastaa tavoiteltuja toimintoja. Kolmantena ongelmana on puutteellinen dokumentointi. Usein ylläpitotoimintoja tekevä henkilö on eri henkilö kuin ohjelman rakentanut, mikä korostaa kunnollisen dokumentoinnin tärkeyttä. [14, p. 464–465]

Ylläpito-ongelmien välttämiseksi olisi ohjelmaa kirjoittaessa kiinnitettävä huomioita ohjelmoijan asiantuntemukseen ja erityisesti hyvien ohjelmointitapojen – dokumentoinnin ja kommentoinnin – ja standardien noudattamiseen. Näin taataan ohjelmointikoodin laatu. Mitä enemmän muutoksia ennakoidaan, sitä helpompia ovat myös ylläpitotoiminnot. [14, p. 459] [16, p. 81] Hyvä tapa helpottaa tulevaisuuden ongelmia on myös ohjelmakoodin vähentäminen, esimerkiksi poistamalla turhaksi jääneitä osioita tai korjaamalla epäselviä rakenteita. [16, p. 81] Käytännössä ylläpito-ongelmia ei voi kuitenkaan täysin välttää, koska teknologinen kehittyminen ja vaatimusten muuttuminen luo jatkuvasti muutospaineita. [14, p. 460] Useimmiten muutokset vaikuttavat ohjelmassa useaan eri osioon. Tällä tavoin virheen korjaus saattaa synnyttää uusia virheitä. Tätä kutsutaan väreilyvaikutukseksi. [16, p. 77]

3.4 Tuoterunkojen ongelmat

Tässä luvussa käydään läpi yleisellä tasolla tuoterunkoarkkitehtuurien teknisiä ja organisaatiotason ongelmia. Tuoterunkoarkkitehtuureihin liittyviä ongelmia – erityisesti muunneltavuuden hallintaa – on käyty läpi myös edellisissä luvuissa. Eri asiakkaiden ja eri järjestelmien tuottamat yksilölliset vaatimukset tuottavat suuria haasteita kattavan rungon tekemiseen. [19, p. 328] Lisähaasteen tuo myös ohjelmistojen kehittyminen suhteessa aikaan ja siten ympäristön vaikutuksesta syntyvät muutospaineet [21, p. 1–2].

Teknisellä puolella suurin haaste on tehdä tuoterunkoarkkitehtuurista riittävän yksinkertainen. Koska tuoteperheet voivat olla hyvin laajoja, syntyy helposti houkutus laajentaa tuoterunkoarkkitehtuurin kohdealuetta ja siten monimutkaistaa ohjelmaa. [13, p. 181–183]. Kuten aiemmin on todettu, ohjelmien monimutkaisuus heikentää huomattavasti tuoterungon laajennettavuutta, ylläpidettävyyttä ja yleistä kehittämistä. Monimutkaisuuteen liittyy myös erilaiset laatuvaatimukset, joita asiakkailta voi olla. Miten sovitetaan yhteen erilaiset suorituskyky-, tarkkuusvaatimukset tai esimerkiksi ohjelman laajuuden vaatimukset? [19, p. 327] Esimerkiksi, jos tuoteperheeseen kuuluu kaksi tuotetta, joista toinen on toisen tuotteen halvempi versio, joka tarvitsee vain osan kalliimman tuotteen ominaisuuksista. Toteutuksessa halvemman tuotteen koodiin jää paljon turhia rakenteita ja käyttämättömiä rajapintoja, mikä monimutkaistaa koodin rakennetta. Halvemmallalla ja kalliimmalla tuotteella voi olla myös erilaiset rautavaatimukset, jotka on otettava ohjelmoinnissa huomioon. [19, p. 328] Koska tuoteperheiden tuotteet ovat keskenään erilaisia, niiden testaaminen on yksi tuoterunkoarkkitehtuurien suurimmista teknisistä haasteista. On hankalaa tehdä kattavaa testaussovellusta, joka kattaisi kaikki tuoteperheen tuotteet. Testaamisesta tulee herkästi työlästä ja myös kallista. Testaaminen hankaloituu sitä mukaan, mitä enemmän joustavuutta eri tuotteiden välillä on. [13, p. 183]

Lisää haasteita syntyy, kun otetaan huomioon kuluva aika. Koska ohjelmat ovat jatkuvassa kehityksessä, eri versioiden komponentit eivät ole välttämättä toistensa kanssa yhteensopivia, mikä vaikeuttaa uudelleenkäyttöä. Komponenttien uudelleenkäyttö on mahdollista vain silloin, kun komponentit tukevat tarvittavia teknologioita ja niiden laatuvaatimukset ovat samalla tasolla kohdejärjestelmän kanssa. [16, p. 269–270]

Usein tuoterunkoarkkitehtuurin liittyvät ongelmat ja haasteet johtuvat kuitenkin muista asioista kuin tekniikasta. Yleisimpiä ongelmien aiheuttajia ovat ihmiset ja organisaatiot. Yhden suuren ongelman aiheuttaa tiedon keskittyminen. Jos arkkitehtuurin suunnittelusta ja toteutuksesta vastaa pieni ryhmä ihmisiä, tieto on vaarassa keskittyä vain pienelle ryhmälle. Toinen vaihtoehto on, että tuoteperheen eri tuotteita kehitetään eri ryhmissä, eikä tieto näiden ryhmien välillä kulje. Joskus projekteihin liittyy niin paljon eri ihmisiä ja instansseja, että tehokas kommunikointi ryhmien välillä tulee käytännössä mahdottomaksi. [19, p. 320] Tähän ongelmaan voidaan vastata panostamalla dokumentointiin ja hyödyntämällä joustavaa organisaatiorakennetta, jonka tarkoituksena on levittää yrityksen sisäistä hiljaista tietoa. [13, p. 181–182] [19, p. 324] Jos tieto arkkitehtuurin toiminnasta on voimakkaasti keskittynyttä ja asiantunteva ihminen ei ole enää organisaation käytettävissä, tulee vastaan suuria haasteita arkkitehtuurin käytössä ja eteenpäin kehittämisessä.

Ongelmaa korostaa myös se, että tuoterunkoarkkitehtuurin suunnittelu ja kehittäminen vaatii useimmiten enemmän taustatietoja kohdeprosessista kuin normaali ohjelmistosuunnittelu, koska tuoterunkoarkkitehtuurin täytyy tukea laajempaa tuoteperhettä. [19, p. 324] Vaadittavan taustatiedon laajuus ja ohjelmiston tuotekehityksen kesto tekee tuoterunkoarkkitehtuurien kehityksestä byrokraattisesti raskaita projekteja verrattuna normaaliin ohjelmistokehitykseen. Vaadittavaa ajankäyttöä taustatietojen etsimi-

seen, komponenttien kehittämiseen ja erilaiseen iterointiin on hankala arvioida. [13, p. 181–183] [19, p. 326]

Tuoterunkoarkkitehtuurin suunnittelu ja käyttöönottoaminen on projekti, joka vaatii organisaatiolta paljon: arkkitehtuurin suunnitteluun on varattava riittävä määrä aikaa ja resursseja. Arkkitehtuurin taloudelliset hyödyt eivät ole välttämättä heti havaittavissa. Jotta tuoterunkoarkkitehtuuri on kannattava yritykselle, yrityksen on oltava valmis sitoutumaan pitkäjänteiseen projektiin. Tämä ei ole nykypäivän yrityksissä itsessään selvää: trendinä on keskittyä lyhyen tähtäimen projekteihin, koska kilpailutilanteet ja markkinat muuttuvat niin nopeasti. Tämä ei tue pitkäjänteisyyden tavoittelua. Usein yritysten johdosta puuttuu tarvittava teknologinen näkemys, jonka avulla tuoterunkoarkkitehtuurin kaltainen projekti saataisiin toteutettua. Yksi syy tähän on hankaluus arvioida ja ennustaa tuoterunkoarkkitehtuurin positiivisia ja negatiivisia puolia. Ennustussmallin täytyisi muistaa ottaa huomioon myös ohjelmistokehityksen tarpeet. [19, p. 320]

4 KIINTEISTÖAUTOMAATIOJÄRJESTELMÄN TUOTTEISTUS

Kiinteistöautomaatiojärjestelmän tuotteistaminen on prosessi, joka vaatii työtä. Onnistuneen tuotteen kehittäminen vaatii idean jalostamista. Tässä luvussa käydään läpi kiinteistöautomaatiojärjestelmän tuotteistusprosessia. Tuotteistusprosessin pohjana toimii luvussa 2 esitetty innovointiprosessimalli. Koska tuoteidea on jo selkeä, tuotteistusprosessi alkaa vaiheesta 2 eli taustatutkimuksesta. Tämän jälkeen siirrytään vaiheen arvioinnin kautta markkinatutkimukseen ja liiketoimintasuunnitelmaan.

4.1 Kiinteistöautomaation taustatutkimus

Tuotteistuksen lähtökohtana on yrityksen idea ja halu liittyä mukaan kiinteistöautomaatiojärjestelmien tuottajaksi. Idean lähtökohtia, ongelmia ja mahdollisuuksia on syytä arvioida sekä yrityksen sisäisten valmiuksien ja mahdollisuuksien kannalta että siltä kannalta mitä yritykseen liittymättömiä paineita ja mahdollisuuksia ideaa kohtaan on olemassa.

4.1.1 Yrityksen sisäisten valmiuksien arviointi innovointiprosessin aloittamiseen

Tärkeimmät pohdittavat asiat yrityksen sisäisistä valmiuksista saattaa tuotteistusprosessi menestyksekkäästi päätökseen ovat seuraavat. Onko yritys sellaisessa tilassa, että se on valmis aloittamaan tuotekehitysprojektin? Tässä tapauksessa yrityksen tila tarkoittaa sen taloudellista vakavaraisuutta, eli taloudellisia mahdollisuuksia aloittaa tuotekehitysprojekti? Taloudellisten valmiuksien tärkeys korostuu, jos projektin toteutus tehdään tuoterunkoarkkitehtuurina, koska tällöin projektin alkukustannukset ja resurssitarpeet ovat suuremmat kuin perinteisessä ohjelmistotuotannossa. Toinen taloudelliseen näkökantaan liittyvä pohdittava asia on suunniteltavan tuotteen tuottopotentiaali. Käytännössä on selvítettävä, onko yrityksellä esimerkiksi valmiita asiakaskontakteja, joille tuotetta on mahdollista myydä. Valmiit kontaktit helpottavat markkinoille pääsemistä.

Taloudellisten näkökulmien lisäksi, on kannattavaa arvioida myös yrityksen oman osaamisen tasoa alalta. Osaamisen puute ei kuitenkaan välttämättä ole toteutukselle kriittinen asia. Riittävät resurssit antavat hyvän mahdollisuuden toteuttaa myös projekteja, joihin ei ole perusteellisia lähtötietoja.

Käytännössä asiakaskontaktien olemassa olon selvitys ja se millaista osaamista yrityksessä on, vastaa kysymykseen sopiiko ideoitu tuote yrityksen tuoteportfolioon. Jos

tuote ei sovi, on pohdittava onko yrityksen osaamisen laajentaminen täysin vieraalle alalle kannattavaa.

Vastaukset kysymyksiin selvitettiin yrityksen sisäisillä haastatteluilla. Haastattelut suoritettiin avoimina haastatteluina, vailla kysymyspohjaa. Avoin haastattelu takaa sen, ettei haastattelussa jäädä ennalta määriteltyihin kysymyksiin. Näin on mahdollista saada syvällisempää tietoa. [22, p. 32] Haastateltaviksi valittiin yrityksen eri toimialojen asiantuntijoita. Haastateltavat on lueteltuna tämän työn liitteessä 1A.

Suunnittelupäällikön mukaan tuotekehitysprojektiin tarvittavien resurssien kannalta yrityksen tilanne on hyvä. Aikaa ja valmiuksia projektin aloittamiseen on. Tuotekehitysprojektia varten on palkattu opinnäytetyön tekijä. Tällä hetkellä yritys palvelee hyvin laajasti teollisuuden ja rakennusalan asiakkaita. Teollisuudelle tuotetaan monipuolisia automaattioratkaisuja. Kiinteistöautomaatioprojektin kannalta kiinnostavaa on yrityksen osaaminen rakennusteollisuuden alalta. Tällä hetkellä asiakkaille tuotetaan muun muassa LVI-, sähkö- ja automaattiosuunnitelmia, joten sisäistä osaamista sekä kiinteistöjen toimilaitteista että automaation tarpeista on tarjolla. Yritys tuottaa kuitenkin vain suunnittelupalveluja. Urakointi tuotetaan alihankkijoilla. [Liite 1A: a]

Projektin kannalta erityisen hyödyllistä on se, että yrityksessä on paljon kiinteistöautomaatioprojektille hyödyllisiä asiantuntijoita. Yrityksessä on automaatio-, rakennustekniikka-, arkkitehtuuri- ja energiatekniikka-ammattilaisia. Kaikilta näiltä ammattiryhmiltä löytyy paljon hyödyllistä tietoa kiinteistöautomaatiojärjestelmien toteuttamiseen. Suunnittelupäällikön mukaan, yrityksessä on pitkään ollut ongelmana yhdistää sisäistä osaamista – teollisuuden ja rakennuspuolen osaajat ovat yrityksessä perinteisesti tehneet vain omia töitään. Samaa sanoivat niin suunnittelupäällikkö kuin rakennussuunnittelija-asiantuntija. Kun kiinteistöautomaatiojärjestelmästä tehdään yritykselle tuote, yrityksen sisäistä saadaan potentiaalia hyödynnettyä. Koska yritys teettää rakennusalan urakoitsijoille kattavia suunnitelmia kiinteistöistä, on hyvin luonnollista, että seuraava askel olisi myös itse alkaa suunnitella kiinteistöautomaatiojärjestelmiä. Näin ollen rakennuspuolen osaajat voisivat tehdä sellaisia automaattiosuunnitelmia, jotka sopisivat hyvin yrityksen omalle kiinteistöautomaatiojärjestelmälle. Koska yritys on jo tunnettu rakennusalan toimijoiden keskuudessa, yrityksellä on käytännössä jo valmiita asiakaskontakteja ja sitä myöten hyvä potentiaali päästä mukaan markkinoille. Yrityksellä on myös valmiita kontakteja kiinteistöautomaatiojärjestelmäasiantuntijoihin. [Liite 1A: a, c]

Motivaatio idean jalostamiseen tuotteeksi on myynnin lisääminen ja luonnollinen laajentuminen vastaamaan yhä suurempaa asiakassegmenttiä. Sisäisen tietotaidon niin automaatio, rakennusalan ja muiden yrityksen edustamien alojen puolelta hyödyntäminen sekä valmiit asiakaskontaktit osoittavat, että kiinteistöautomaatiojärjestelmä olisi yrityksen kannalta sopiva uusi tuote tuoteportfolioon.

4.1.2 Kiinteistöautomaatiomarkkinoihin vaikuttavat paineet

Ulkoisilla paineilla tarkoitetaan tässä yhteydessä sitä millaisia muutospaineita kiinteistöautomaatiojärjestelmien toimittajat alalla kohtaavat. Asia voidaan karkeasti jakaa kahteen osaan: alan kilpailu ja lakimuutokset.

Ensimmäisenä on syytä käsitellä kiinteistöautomaatioalan kilpailutilanne. Selvitys tehtiin tutustumalla kiinteistöautomaatiojärjestelmiä suunnittelevien yritysten Internet-sivuihin ja mainosmateriaaleihin. Taulukossa 4.1 on esitetty tutkimukseen mukaan otetut yritykset. Yritykset ovat valittu mukaan asiantuntijahaastattelujen ja Internet-hakujen perusteella.

Taulukko 4.1: Suomessa toimivia kiinteistöautomaatiojärjestelmiä suunnittelevia yrityksiä. Sivustot tutkittu 1.2.2017.

Yritys	Internetsivut
Fidelix	https://www.fidelix.fi/
Beckhoff Oy	http://www.taloautomaatio-beckhoff.fi/
Ouman Oy	http://ouman.fi/palvelut/rakennusautomaatio/
Honeywell	https://www.honeywell.com/worldwide/emea/finland-fin
Siemens	http://www.siemens.fi/fi/infrastructure_and_cities/talotekniikka.htm

Taulukossa listatut yritykset vaikuttavat kaikkialla Suomessa. Yritysten Internet-sivustojen avulla pyritään selvittämään, millaisia rakennusautomaatiotuotteita yrityksillä on tarjota, eli käytännössä mitä mainosmateriaalit lupaavat tuotteiden toteuttavan. Eri-tyisesti pyritään selvittämään mitä markkinoilla jo olevat järjestelmät lupaavat asiakkaille. Mainosmateriaaleista saadut tulokset on koottu taulukkoon 4.2. Käytännössä taulukko on rakennettu etsimällä mainosmateriaaleista avainsanoja ja lupauksia toiminnoista.

Taulukko 4.2: Suomessa toimivien kiinteistöautomaatiojärjestelmiä toimittavien yritysten kiinteistöautomaatiotuotteiden ominaisuuksien vertailua. Järjestelmät lupaavat hyvin samankaltaisia asioita.

	Kiinteistön toimintojen hallinta	Skaalautuvuus	Energia-tehokkuus	Kustannussäästöt
Fidelix	X	X	X	X
Beckhoff Oy	X	X	X	X
Ouman Oy	X		X	X
Honeywell	X			
Siemens	X		X	X

Taulukon toinen sarake, kiinteistön toimintojen hallinta, tarkoittaa rakennusten eri osien – esimerkiksi ilmanvaihdon, lämmitysjärjestelmän ja niin edelleen – hallintaa. Kaikki toimittajat lupasivat järjestelmänsä pystyvän hallitsemaan näitä toimintoja, joten taulu-

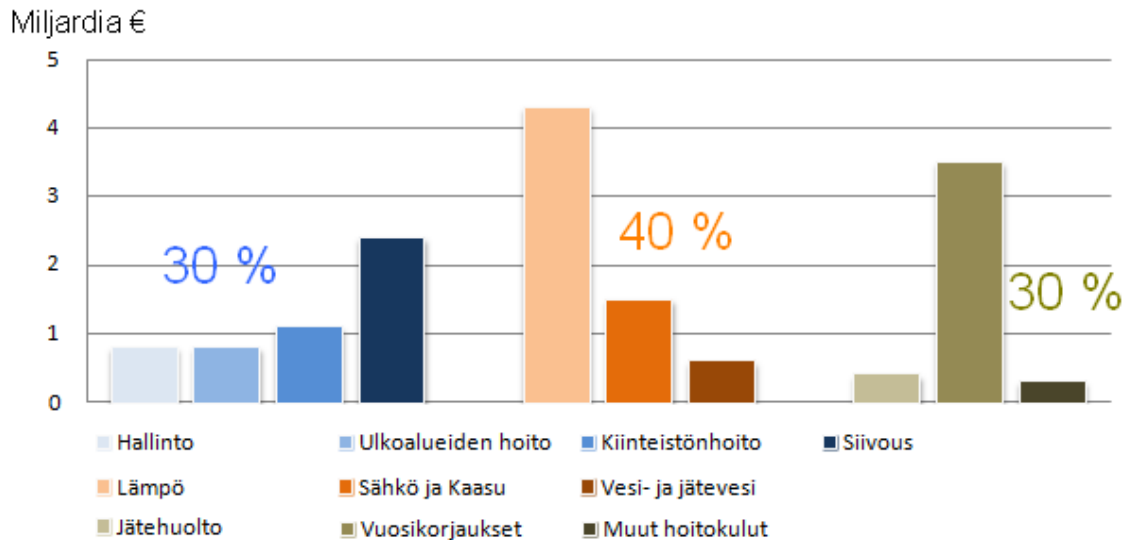
kossa 4.2 tämä asia on tiivistetty yhteen sarakkeeseen. Skaalautuvuus tarkoittaa tuotteen mahdollisuuksia toimia erikokoisissa rakennuksissa. Energiatehokkuus viittaa tuotteen ympäristöystävällisyyteen ja kustannustehokkuus siihen, että tuotteen käyttöönottonen on säästää asiakkaan rahaa tulevaisuudessa esimerkiksi juuri energiansäästön kautta. Taulukosta nähdään, etteivät tuotteet eroa ominaisuuksiltaan toisistaan paljon. Kaikki tuotteet lupaavat rakennuksen ominaisuuksien hallintaa. Lisäksi lähes kaikki tuotteet lupaavat energiatehokkuutta ja kustannussäästöjä. Ainoastaan Honeywellin tuotteella oli niin suppeat mainosmateriaalit, että niistä kävi selväksi ainoastaan toimintojen hallinta.

Mainos- ja Internet-materiaaleista ei selviä kuinka eri tuottajien ratkaisut eroavat teknisesti toisistaan. Tutkimuksen tässä vaiheessa sillä ei kuitenkaan ole merkitystä, koska tarkoituksena on selvittää, miten tuotteet näyttäytyvät asiakkaille. Koska tuotteet ovat ulkoisesti samankaltaisia, voidaan suurimman kilpailuvaltin olettaa olevan hinta. Hintakilpailun ongelma on se, ettei kiinteistöautomaatio ole kovin arvokasta. Sen osuus rakennuskustannuksista on 1–1,5 %, siis hyvin vähän verrattuna kokonaiskustannuksiin. Asiakkaalle järjestelmäkustannusten takaisinmaksuaika on potentiaalisesti lyhyt, jonka luulisi olevan hyvä houkutin markkinoilla. [3, p. 12]

Tiukka taloudellinen kilpailu ja järjestelmien samankaltaisuus tuo haasteita yritykselle, joka aikoo liittyä uutena tekijänä mukaan markkinoille. Tärkeää on pyrkiä löytämään keino, millä erottautua markkinoiden muista toimijoista. Toisaalta markkinoiden yhtenäisyys myös helpottaa markkinoille pääsemistä, koska on hyvin selkeää, millaiset käytännöt alalla vallitsevat. Jotta markkinoille on mahdollista päästä, järjestelmässä on oltava olemassa tiettyjä perusominaisuuksia. [23, p. 44]

Kilpailijoiden kartoituksen lisäksi on syytä pohtia, millaiset tulevaisuuden näkymät alalla ovat. Onko alalla tulevaisuutta? Lähestymiskulma tähän kysymykseen on tutkia, millaisia tuloksia kiinteistöautomaatiojärjestelmillä on saatu aikaan. Mitä merkittävämpiä tuloksia on mahdollista saavuttaa sitä todennäköisemmin alalla on tulevaisuutta. Toinen näkökulma tulevaisuuden haasteisiin on hallinnolliset ja valtiolliset toimenpiteet. Onko esimerkiksi ilmastositomuksilla vaikutusta kiinteistöautomaation tulevaisuuden näkymiin?

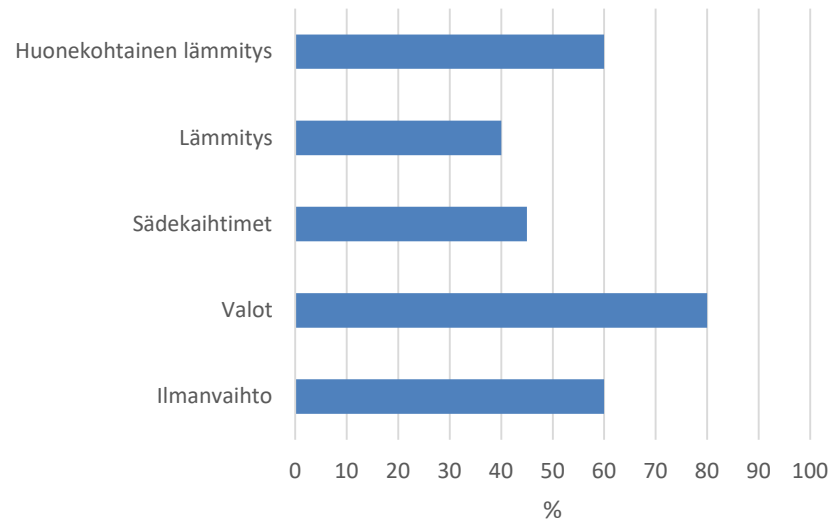
Kiinteistöautomaatiota käytetään mukavuuden, turvallisuuden ja energiatehokkuuden lisäämiseen kaikenlaisissa rakennuksissa. Suomessa on jo pidempään vaadittu muuhun Eurooppaan nähden verrattain tehokasta lämmöneristystä. Syy on Suomen haasteellinen maantieteellinen sijainti: vuodenaikojen välinen lämpötilavaihtelu on suuri. Rakennusmateriaaleilla voidaan vaikuttaa siihen, miten paljon rakennus luovuttaa lämpöenergiaa. Suomen kaikesta energiankulutuksesta 40 % kuluu kiinteistöjen tarpeisiin. [24, p. 13] Suomen kiinteistöt tuottavat 30 % Suomen hiilidioksidin päästöistä [25]. Pääasiassa energiaa kuluu lämmitykseen ja jäähdytykseen. Kuvassa 4.1 on Ympäristöministeriön teettämä taulukko, jossa näkyy miten suomalaisen rakennuskannan ylläpidon kustannukset jakaantuvat. [1, p. 13]



Kuva 4.1 Suomalaisen rakennuskannan ylläpidon kustannusten jakaantuminen. Suurin kustannus aiheutuu lämmityksestä. Mukailtu lähteestä [24, p. 2]

Kuvasta voidaan todeta, että rakennusten lämmitys on edellä mainitun energiankulutuksen lisäksi myös suurin kuluerä suomalaisessa rakennuskannan ylläpidossa. Yhteensä rakennuskannan kustannukset ovat 15,5 miljardia euroa. Suurin vaikutus energian kulutukseen on Suomen pohjoinen sijainti. Suuret lämpötilavaihtelut vuoden sisällä tekevät kiinteistöjen energiatehokkaasta lämmityksen hallinnasta hyvin haastavaa. Kesällä asuntoja pitäisi saada jäähdytettyä tehokkaasti ja talvella taas lämmitettyä. Maantieteellisten olosuhteiden vuoksi on odotettavissa, että lämmityksen tarve ja kulut tulevat pysymään korkealla. Lämmityksen tarpeeseen voidaan vaikuttaa paremmilla rakennusmateriaaleilla ja -tekniikoilla. Esimerkiksi vaihtamalla ikkunat energiatehokkaampaan malliin saadaan aikaan jo paljon hyötyä. Lämmityksen ja jäähdytyksen ongelmaan voidaan kuitenkin fyysisten ratkaisujen lisäksi vaikuttaa automaatiota ja säätötekniikkaa lisäämällä. [24, p. 13]

Automaation vaikutuksesta energian kulutukseen on tutkittu muun muassa Saksassa. Kuvassa 4.2 on esitetty KNX Associationin mainostama energiansäästö kiinteistöissä.



Kuva 4.2 Energian kulutuksen pieneneminen automatisoinnin ansiosta. Mukailtu lähteestä [26].

Tutkimuksen mukaan suurimmat hyödyt automatisoinnista saadaan valojen hallinnasta. Myös muissa automatisoinnin osa-alueissa on nähtävissä selkeitä hyötyjä. Suomen kiinteistökannan ongelmien kannalta on merkittävää, että hyötyjä on nähtävissä myös lämmityksen hallinnassa. Olosuhteiden hallinnalla on myös vaikutusta kiinteistön arvoon. Jos ohjaukset on suoritettu toimivalla tavalla, ne voivat nostaa kiinteistön arvoa jopa 10 %. [27]

On kuitenkin olemassa myös tutkimuksia, joiden mukaan lämmityksen automatisoinnilla ei ole ollut juurikaan vaikutusta rakennusten energiankulutukseen. Bremen Energie-Consens ja Bremen Energy Institute järjestivät kolmen rakennuksen kokeen, jossa jokaiseen rakennukseen asennettiin keskitetty lämmityksenhallintajärjestelmä. Tuloksia verrattiin kolmeen samanlaiseen rakennukseen, joiden lämmitystä hallittiin perinteisillä elektronisilla järjestelmillä. Uusilla järjestelmillä ei ollut käytännössä vaikutusta energian kulutukseen. Syy tähän ei kuitenkaan ollut varsinaisesti automaatioissa. Tutkimuksen mukaan asukkaat ohjasivat jo valmiiksi rakennuksia energiatehokkaasti manuaalisesti termostaatteja säätäen, joten automaation tulokset eivät olleet kovin näkyviä. Tämä on merkki siitä, että automatisoitujärjestelmä oli liian vaikeakäyttöinen – uuden järjestelmän käyttämistä ei jaksettu opetella. Tutkimus on osoitus, että tietyissä tilanteissa myös manuaalisella säädöllä voidaan päästä energiatehokkaaseen ratkaisuun. Yhtä korkeaan käyttömukavuuteen sillä tuskin kuitenkaan päästään, sillä manuaalinen säätö vaatii huomattavasti enemmän työtä käyttäjältä. Tutkimuksen tärkein osoitus on, ettei kehittynytkään järjestelmä ole toimiva, jos se on liian vaikeakäyttöinen. [28]

Edelliset kappaleet osoittavat, että kiinteistöautomaatiojärjestelmistä on hyötyä. Löydetty tutkimukset käsittelivät selkeästi eniten energiatehokkuuden parantumista. Todennäköinen syy tähän on, että energiatehokkuus on ilmiselvä keino luoda säästöjä. Kiinteistöautomaation tuomat muut edut kuten mukavuus, eivät ole yhtä helposti mitattavissa kuin energiansäästö- ja tehokkuus.

Energiatehokkuuden painotuksen takana ei ole pelkästään taloudelliset näkökulmat vaan myös ympäristöystävällisyys. On siis hyvin luonnollista, että myös kiinteistöautomaatiojärjestelmien mahdollisuuksia energiatehokkuudessa tutkitaan myös ympäristöystävällisyyden näkökannalta.

Käytännössä ympäristöystävällisyys näkyy Suomessa ympäristötavoitteina. Tavoitteita syntyy muun muassa kansainvälisistä ilmastopöytäkirjoista. Suomi on esimerkiksi mukana Kioton sopimuksessa, jonka tavoitteena on laskea kasvihuonepäästöt vuoden 1990 tasolle. Euroopan Unionin (EU) toimenpideohjelmassa on 20 % energiansäästötavoite vuoteen 2020 mennessä. Keskeinen tapa, jolla tavoitteisiin aiotaan pyrkiä, on rakennusten energiatehokkuuden parantaminen. Ympäristöministeriön mukaan ensimmäinen askel on rakennusten häviöiden pienentäminen, mutta toisena keskeisenä asiana on kiinteistöautomaation lisääminen: laitteiden ohjauksen ja valvonnan tehostaminen. [1, p. 3–7] Toinen selkeästi kiinteistöautomaation tarpeeseen liittyvä lakimuutos on lähes nollaenergiataloja koskeva EU-direktiivi. Sen mukaan vuoden 2012 alusta lähtien kaikkien uusien rakennusten tulee olla lähes nollaenergiataloja, eli taloja, jotka tuottavat uusiutuvia energiavaroja yhtä paljon kuin kuluttavat. [29]

Rakennuksen automaation tehokkuusstandardi SFS-EN 15232 määrittää hyvin pitkälti, minkälaista automaatiota suomalaisiin rakennuksiin pyritään tuottamaan. Standardi jakaa tehokkuuden neljään eri luokkaan, joita vastaavat erilaiset automaatiotasot. Taulukossa 4.1 on esitetty tehokkuusstandardin mukainen luokkajako. [1, p. 8–9]

Taulukko 4.1: *Rakennuksen automaation tehokkuusstandardin mukainen luokkajako.* [1, p. 8–9]

Luokka	Automaatiotaso
A	Talotekniikan hallintajärjestelmä. Luokan B lisäksi raportointi ja vika-diagnoosi.
B	Rakennuksen automaatiojärjestelmä. Luokan C lisäksi kommunikoivat huonesäätimet.
C	Automaattiset säätö- ja ohjaustoimet. Esimerkiksi käsikäyttöinen valokytkin ja automaattinen sammutus sekä automaattiset huonesäätimet
D	Manuaalinen käyttö. Esimerkiksi käsikäyttöinen valokytkin ja patteriventtiili.

Mitä korkeammalle automaatiotasolle päästään, sitä automatisoidumpi rakennus on. Teknologia ei rajoita korkeimmalle automaatiotasolle pääsemistä, mutta yleisesti sen saavuttamiseen on vielä matkaan. A-luokka tarjoaisi kiinteistön omistajalle keinoja seurata ja ohjata kiinteistön energiankulutusta ja käyttöä. Luokka C on minimivaatimus uudisrakentamiselle ja B on suositusluokka. Jokaiselle automaatiotasolle on standardissa määritelty tarkasti omat vaatimukset. [1, p. 10]

Edellä olevat direktiiveihin, standardeihin ja ilmastotavoitteisiin liittyvät esimerkit paljastavat sen, että kiinteistöautomaatiolla on tulevaisuutta myös yhteiskunnalliselta näkökannalta. Suunnitelmia on tehty valtiollisella tasolla asti. Erityisen voimakkaalta

tavoitteelta vaikuttaa energiatehokkuuden lisääminen. Syy on selkeä: energiatehokkuudella voidaan tehokkaasti tuottaa säästöjä ja vaikuttaa ympäristökuorman määrään. Toiseksi, kiinteistöautomaatiojärjestelmät ovat erityisen tehokkaita nimenomaan energiatehokkuuden parantamisessa.

Kiinteistöautomaatiojärjestelmien yleistymisessä on kuitenkin olemassa myös haasteita. Insinööritieteiden ulkopuolella ei välttämättä ymmärretä kuinka suuria etuja automaatiojärjestelmät voivat tuottaa. Satu Marttila on käsitellyt diplomityössään *Home automation – A challenge for electrical desingers, contractors and electricians* (TKK, 2009) miten kotiautomaatioon liittyvät viiteryhmät – urakoitsijat, kiinteistöjen omistajat ja suunnittelijat – kokevat automaation. Vaikka Marttilan työ käsittelee kyselytutkimuksen avulla kotiautomaatiota ja työ painottuu suuremman mittaluokan kiinteistöautomaatioon, työn tuloksia voidaan analysoida myös kiinteistöautomaation viitekehityksessä. Myös Joonas Ojalan insinöörityö *Kiinteistöautomaatiojärjestelmien käyttäjänäkökulmaisen kehittäminen* (Haga-Helia, 2015) antaa näkökulmia siihen, millaisia ongelmia kiinteistöautomaatiojärjestelmissä on.

Suurin ongelma automaatiojärjestelmien yleistymisessä on heikko markkinointi: asiakkailta on liian vähän tietoa, mitä etuja automaatiojärjestelmä voi tuottaa. Automaatiota ei asiakaspuolella yksinkertaisesti nähdä tarpeellisenä. Huono markkinointi näkyy niin asiakkaiden suhtautumisessa kuin muidenkin viiteryhmien suhtautumisessa. Asiakkailta on helposti käsitys automaatiosta luksustuotteena, johon ei välttämättä ole varaa. Kun asiakkaat pitävät automaatiota luksustuotteena syntyy herkästi epärealistisia odotuksia. Rakennusala itsessään on melko konservatiivinen ala ja vaikka rakennuksia on automatisoitu jo hyvin kauan, automaation yleistyminen kohtaa jonkinlaista vastarintaa. Koulutuksen puute on todennäköisin syy vastarinnalle. Viiteryhmien välillä tarvitaan lisää yhteistyötä, jotta automaatiojärjestelmien suosio kasvaisi. [30, p. 53] [23, p. 45–55]

Edellisen kappaleen väitteet kohdistuvat siis erityisesti kotiautomaatioon. Kuilu kiinteistöautomaation ja kotiautomaation välillä tuskin kuitenkaan on kovin suuri, vaikka kiinteistöautomaation puolella on totuttu paljon suurempaan automaatiotasoon. Suurimmat ongelmat myös kiinteistöautomaation puolella liittynevät markkinointiin ja kiinteistön rakentamiseen liittyvien viiteryhmien yhteistyökykyyn.

4.1.3 Taustatutkimuksen johtopäätökset

Tämä luku vastaa käytettävän innovointiprosessimallin porttia 2, eli taustatutkimuksen tulosten arviointia. Yrityksen omat valmiudet kiinteistöautomaation tuotekehitysprojektiin mukaan lähtemiseen ovat hyvät: yrityksessä on riittävästi tietotaitoa, valmiita asiakassuhteita ja resursseja projektin aloittamiseen. Erityisen tärkeää on se miten hyvin kyseinen tuote sopii yrityksen tuoteportfolioon. Omien valmiuksien lisäksi kiinteistöautomaation mahdollisuudet vaikuttavat lupaavilta: järjestelmillä on mahdollista vaikuttaa kiinteistöjen energianhallintaan ja -kulutukseen sekä muihin toimintoihin. Lisäksi lainsäädännöllisten muutosten vuoksi voidaan olettaa, että kiinteistöautomaation tarve tulee

tulevaisuudessa kasvamaan. Lupaavien tulosten johdosta tuotteistamisessa siirrytään seuraavaan vaiheeseen.

4.2 Markkinatutkimus

Taustatutkimuksen jälkeen innovointiprosessissa siirrytään vaiheeseen 3 eli liiketoimintasuunnitelman tekemiseen. Käytännössä tämä tarkoittaa markkinoiden tuntemuksen syventämistä. Tärkeintä on saada tietoa siitä, millainen on eri käyttäjäryhmien mukaan hyvä järjestelmä ja millaisia ongelmia nykyisissä järjestelmissä on. Kun näihin asioihin saadaan vastaus, on mahdollista pohtia kilpailunäkökulmia.

4.2.1 Millainen on hyvä kiinteistöautomaatiojärjestelmä?

Jotta voidaan suunnitella markkinoille sopiva kiinteistöautomaatiojärjestelmä, on pohdittava millainen on hyvä järjestelmä. Taustatutkimuksen ja luvun 1 ansioista tiedetään, että kiinteistöautomaatiojärjestelmän tulee käsittää kokonaisvaltaisesti rakennuksen hallinta: järjestelmän pitää vaikuttaa positiivisesti niin energian hallintaan, turvallisuuteen kuin mukavuuteen. Hyvän järjestelmän ominaisuuksien pohtimisessa voidaan jälleen hyödyntää Marttilan diplomityössä ja Ojalan insinööritöydessä toteutettuja kyselytutkimuksia, soveltaen tuloksia koskettamaan kiinteistöautomaatiojärjestelmiä.

Kuten luvussa 4.1.2 mainittiin, automaatiojärjestelmien markkinoinnissa on selkeitä puutteita. Tärkeä kysymys onkin, miksi asiakas tahtoi kiinteistöautomaatiojärjestelmän? Marttilan kyselytutkimuksen mukaan urakoitsijoiden ja suunnittelijoiden mielestä asiakkaalle tärkein ominaisuus on mukavuuden lisääminen. Energiansäästö on Marttilan kyselyssä viiteryhmien mielestä asiakkaalle keskivertotärkeä. Suurin ongelma oli, ettei asiakkailla ollut riittävästi informaatiota järjestelmistä. [30, p. 51] Suureksi osaksi tiedon puutteen vuoksi asiakkaat kokivat, etteivät ole kotiautomaation kohderyhmää. Mainostamisen ja markkinoinnin tehokkuudesta kertoo se, että jos järjestelmässä on olemassa jokin erikoinen, mutta selvästi nähtävissä oleva ominaisuus, esimerkiksi keittiön valo vilkkuu vesivahingon merkiksi, saa kyseinen ominaisuus erinomaista palautetta. [30, p. 58] Asiakkaalle on siis pystyttävä osoittamaan tuotteen toimivuus. Jokin selkeästi esillä oleva ominaisuus voi hyvin olla markkinointia varten tehty. Mahdollisesti yksi ongelma kiinteistöautomaatiojärjestelmän suosion kasvattamisessa on, että hyvän järjestelmän toimintaa ei ideaalitulanteessa huomaa.

Ojalan kyselytutkimuksessa niin urakoitsijat, asentajat kuin järjestelmän omistajat luettelivat kiinteistöautomaatiojärjestelmille asetettuja vaatimuksia. Urakoitsijoiden, asentajien ja omistajapuolen vaatimukset ovat hyvin samankaltaisia: järjestelmän pitää olla helppokäyttöinen ja sisältää graafinen suomenkielinen käyttöliittymä. Mitä helppokäyttöisempi järjestelmä on, sitä tehokkaampaa sen käyttäminen on. Järjestelmän on myös oltava mahdollisimman monikäyttöinen eli integroitavissa useaan eri järjestelmään. Käytännössä tämä tarkoittaa sitä, että järjestelmän olisi hyvä tukea mahdollisimman montaa eri kenttäväylää. Siten toimilaitteiden integroitavuus on helppo toteuttaa. Myös etäkäytön mahdollisuudet on nykyään syytä ottaa huomioon. Viimeisin vaatimus

liittyy järjestelmän käyttämiseen ja huoltoon: huollon tarpeen tulisi olla matala eli toimintavarmuuden korkea. Jos huollolle kuitenkin on tarvetta, tulisi sekä huollon että varaosien olla helposti saatavilla, mikä pidentää järjestelmän elinkaarta. [23, p. 16–18]

Kiinteistöautomaatiojärjestelmille asetetut vaatimukset ovat kaiken kaikkiaan yleistajuisia ja selkeitä. Erityisen tärkeää on toimintavarmuus ja helppokäyttöisyys. Tällaiset vaatimukset ovat osoitus siitä, että rakennusosalalla ovat selkeät käytännöt miten asioita hoidetaan. Alan konservatiivisuus nousee hyvin esille. Kokonaisuutena kiinteistöautomaatiojärjestelmän vaatimukset voidaan jakaa toiminnallisiin, teknisiin ja taloudellisiin. Toiminnalliset ominaisuudet tarkoittavat helppokäyttöisyyttä, käyttäjän vuorovaikutusta kiinteistön kanssa, tilojen ratkaisujen toimivuutta, turvallisuusnäkökulmia ja varmatoimisuutta. Tekniset ominaisuudet taas tarkoittavat sitä, miten järjestelmä on hallittavissa ja säädettävissä: ovatko eri osiot säädettävissä itsenäisesti, yhdessä, yksilöllisesti tai muulla tavalla. Tekniset ominaisuudet vastaavat myös rautatason ominaisuuksista, eli siitä miten hyvin järjestelmä on laajennettavissa, huollettavissa, muutettavissa tai liitettävissä muihin järjestelmiin kuten kunnalliseen kiinteistöautomaatiovalvomoon. Yhteensopivuus voi olla toimilaitetason tai automaatiotason ongelma. Viimeinen arvioitava ominaisuus on järjestelmän taloudelliset ominaisuudet. Järjestelmän tulee olla kustannustehokas ja sisältää raportointiin soveltuvia työkaluja. Mitä paremmin järjestelmä pysyy kunnossa ja päivitettävissä sitä arvokkaampi se on elinkaarensa aikana. [5]

4.2.2 Kiinteistöautomaation tulevaisuuden suuntaukset

Taustatutkimuksessa käsiteltiin kiinteistöautomaatioalan yleisiä näkymiä. On kuitenkin syytä myös tarkemmin pohtia, mihin suuntaan alan kehitys on menossa. Tulevaisuudessa automaation merkitys rakentamisessa on kasvava. Tämä johtuu edellä mainituista lainsäädännöllisistä vaikuttimista, automaation halpenemisesta että automaatiotietoisuuden lisääntymisestä. Erityisesti mahdollisuuksia on suuremmissa kohteissa kuten sairaaloissa. Alalla on selkeää tarvetta paremmalle tuotteistamiselle ja markkinoinnille. Pelkästään markkinointi ei kuitenkaan riitä, vaan tietoisuutta on pyrittävä kasvattamaan myös huoltomiesten ja urakoitsijoiden puolella. Tällä tavoin on kokonaisvaltaisesti mahdollista vaikuttaa kiinteistöautomaation suuntauksiin. [30, p. 65]

Teknisesti alalla on selkeä suuntaus kohti älytaloa ja standardin SFS-EN 15232 mukaista A luokan automaatiotasoa. Odotettavissa on yhä kattavampia raportointityökaluja kiinteistön kunnosta, huollon tarpeesta ja energiankulutuksesta. Eri raporteilla voi olla eri kohderyhmä: kiinteistöjen omistajille, huoltomiehille, valvomotyöntekijöille kuin käyttäjille voi olla tarkoituksenmukaiset raporttinsa. Mitä enemmän tietoa järjestelmästä on saatavissa, sitä helpompaa sitä on hallita ja tehdä faktoihin perustuvia ratkaisuja. Tieto pitää saada mukaan päätöksentekoon. [23, p. 52] Raporttien mahdollisuudet tulevat parhaiten esiin, kun otetaan huomioon digitalisoinnin tarjoamat mahdollisuudet. Eli myös kiinteistöautomaatiojärjestelmien parissa voidaan hyödyntää yhä enenevässä määrin datan analysointia ja esineiden Internetiä. [31, p. 18]

Yhteiskunta muuttuu yhä kiihtyvämpään tahtiin, mikä tuottaa haasteita pysyville rakenteille. Tulevaisuuden järjestelmät tulevatkin olemaan myös monikäyttöisempiä ja muuntautuvampia kuin edeltäjänsä. On siis pyrittävä varautumaan kiinteistöjen käyttö-tarkoituksen muuttumiseen. [31, p. 18]

Yrityksen sisäiset asiantuntijat – suunnittelupäällikkö, energia-alan asiantuntija ja rakennussuunnittelija – olivat haastateltaessa sitä mieltä, että suurimmat kehityskohdat kiinteistöautomaation parissa ovat käytännössä energiatehokkuus, tarpeenmukainen säätö, etäkäyttö ja langattomuus. Energiatehokkuuden asettamat muutospaineet on käsitelty jo aiemmissa luvuissa. Tarpeenmukainen säätö tarkoittaa sitä onko järjestelmä aktiivinen silloin, kun sitä ei akuutisti tarvita. Tarpeenmukaisen säädön vuoksi esimerkiksi hiilidioksidianturien käyttö on huomattavasti lisääntynyt. Kyseisillä antureilla on mahdollista toteuttaa läsnäoloon perustuvia ohjauksia: kun huone on tyhjillään, on automaatiojärjestelmä lepotilassa. Myös järjestelmän etäkäyttö on asia, joka tulevaisuudessa tulee olemaan yhä tärkeämmässä asemassa. Etäkäyttö on toki nykyäänkin pitkälle kehittynyttä: Internetin välityksellä kiinteistöautomaatiota on mahdollista ohjata käytännössä mistä päin maailmaa tahansa. Tulevaisuuden haaste on mobiililaitteiden ja tablettien hyödyntäminen tehokkaasti. Pienempi näyttö ja toisenlainen järjestelmäarkkitehtuuri verrattuna esimerkiksi tavalliseen tietokoneeseen, tuovat omat haasteensa sekä käyttöliittymien kehittämiseen että integrointiin. Maailma on muuttumassa yhä mobiilimmaksi, joten mobiililaitteiden hyödyntäminen kiinteistöautomaatiojärjestelmien ohjaamisessa on luonnollinen edistysaskel. [Liite 1A:a, b, c]

Toimilaitetasolla kehitys on myös suuntautumassa kohti langattomia järjestelmiä. Erityisesti langaton anturitekniikka on kiinteistöautomaatiojärjestelmien tulevaisuutta. Langattomuus mahdollistaa anturien helpon asentamisen hankaliinkin paikkoihin. Tekniikan avulla on mahdollista säästää valtava määrä johtoa sekä mahdollistaa anturien paikkojen nopean vaihtamisen. Näin voidaan helposti rakentaa uusia tarvittavia mittauspisteitä ja järjestelmiä. Langattoman anturitekniikan edut tulevat tulevaisuudessa esiin erityisesti korjausrakentamisessa. [30, p. 21–23 ja 48]

Samoin kuin monessa muussakin kiinteistöautomaatioon liittyvässä asiassa, myös langattoman anturitekniikan ongelmana on standardien puute: eri valmistajilla on siis omat standardinsa, mikä tekee järjestelmien joustavasta integroinnista hankalaa. Myöskään akkuteknologia ei kaikissa käyttökohteissa ole vielä riittävällä tasolla langattoman anturoinnin hyödyntämiseen ja langattomat anturit ovat hinnaltaan kalliimpia. [30, p. 48]

Kiinteistöautomaation tulevaisuuteen liittyy monenlaisia haasteita, mutta myös uusien teknologioiden ja tekniikoiden luomia mahdollisuuksia. Yrityksen asiantuntijoiden ja kirjallisuuden perusteella voidaan päätellä, että langattomuus tulee olemaan tärkeässä asemassa tulevaisuudessa, niin etäkäytön kuin anturoinnin puolella ja järjestelmän tuottaman tietomäärän hyödyntämisessä.

4.3 Toimialakohtainen haastattelututkimus kiinteistöautomaatiosta

Jotta markkinatutkimuksen tulos ei jäisi ainoastaan kirjallisuusselvityksen varaan, kiinteistöautomaation nykyisiä trendejä selvitetään myös haastattelututkimuksella. Tämä haastattelututkimus on erillinen kokonaisuus yrityksen sisällä tehdyistä haastatteluista, joita käytettiin lähteenä edellä. Haastattelututkimuksen tarkoituksena on tuoda esiin kiinteistöautomaation loppukäyttäjien mielipiteitä. Onnistunut kehitys tarvitsee kontaktin myös loppukäyttäjään, jota haastattelututkimus simuloi. Tutkimuksen tarkoituksena on täydentää aiemmassa markkinatutkimuksessa tehtyjä havaintoja kiinteistöautomaatiojärjestelmistä, mutta ennen kaikkea pyrkiä selvittämään mitä kehityskohteita nykyisissä järjestelmissä on – kiinteistöautomaatiojärjestelmän perustyyppi on hyvin selvillä aiemman tutkimuksen ansiosta.

Haastattelututkimus on laadullinen tutkimusmenetelmä, eli se ei anna suoria mitattavia suureita toisin kuin esimerkiksi Marttilan määrällistämenetelmää lomakehaastattelua hyödyntävä kyselytutkimus. [32] Haastattelututkimuksessa on käytännössä mahdollista valita kahden menetelmän väliltä: teema- ja avoimen haastattelun. Teema-haastattelussa on aina kysymysrunko, jonka avulla haastattelija ohjaa tutkimuksen etenemistä. Avoimessa haastattelussa ei ole runkoa vaan muoto on täysin vapaa. Koska tässä tapauksessa haastattelija on kokematon ja haastatteluilla tahdotaan vastauksia melko tarkkoihin ongelmiin, on teemahaastattelu parempi vaihtoehto. Haastattelut tarjoavat kuitenkin tarvittaessa myös mahdollisuuden poiketa kysymysalueesta. [22, p. 2]

Lomakehaastattelun avulla saadaan aihe rajattua hyvin tarkasti, tehokkaasti ja haastateltua suurta osallistujajoukkoa. Lomahaastattelulla saadun aineiston analysointi ei ole kovin työlästä, koska analysoinnissa on mahdollista hyödyntää esimerkiksi matemaattisia analysointimenetelmiä. Lomakehaastatteluilla saatu informaatio jää kuitenkin usein pintapuoliseksi. Teemahaastatteluiden avulla on mahdollista saada selvitettyä syvällisempää tietoa tutkimuskohteesta kuin lomakehaastattelulla. Syvällisempi informaatio on avoimemman haastattelumuodon lisäksi toinen tärkeä syy, miksi tässä tutkimuksessa on päädytty teemahaastattelumetodiin. Negatiivinen puoli teemahaastattelussa on kuitenkin haastatteluiden analysointi, joka on työläämpää kuin lomakehaastattelun analyysi. Laadullisissa metodissa tulkinnot perustuvat enemmän haastattelijan päätelmiin. Analysoinnin työläyden vuoksi teemahaastatteluissa on käytännössä aina vähemmän haastateltavia kuin lomakehaastatteluissa. [22, p. 2]

Kolmas syy valita tutkimusmetodiksi haastattelututkimus on se, että aiemmin tehdyissä tutkimuksissa haastattelut on hoidettu lomakehaastatteluilla. Teemahaastattelut tuovat aiempien tutkimusten tarjoamaan informaation uutta, syvällisempään, näkökulmaa. Aiemman tutkimuksen metodeja on turha toistaa, kun muitakin on käytettävissä.

Haastattelun kohteiksi valikoidaan viisi projektin suunnittelua tukevaa kohdetta. Kohteiksi pyritään valitsemaan mahdollisimman erilaisia rakennuksia: uusia, vanhoja, suuria ja pieniä. Kokoa ja ikää merkittävämpi jaotteluperuste on kuitenkin rakennuksen

käyttötarkoitus, koska sillä on enemmän merkitystä kiinteistöautomaation kannalta. Tutkimukseen valikoitiin

- toimistorakennus
- koulu
- asuinrakennuksia
- tehdashalli
- juhlatalo.

Toinen tutkimukselle tärkeä jaottelu on haasteltavan henkilön viiteryhmä kiinteistöön nähden. Tarkoituksena on saada eri käyttäjien mielipiteet ja mahdolliset kehityskohteet esiin. Tutkimukseen valikoitiin kiinteistöautomaatiojärjestelmän valvomotyöntekijöitä, kiinteistönhuoltomiehiä ja kiinteistön omistaja. Haastateltavien valinnassa tavoitteena on tuottaa aineisto, jonka avulla tutkittavaa ilmiötä voidaan analysoida. Tämän vuoksi haastateltaviksi on valittu ennen kaikkea teknisen osaamisen ammattilaisia: järjestelmän valvomotyöntekijöitä ja huoltomiehiä, koska he ovat järjestelmän pääasiallisia käyttäjiä ja todennäköisesti tietävät järjestelmistä eniten. Jos haastateltavien joukko koostuu ainoastaan teknisenpuolen ammattilaisista, on uhkana, että tutkimus jää yksipuoliseksi. Sen vuoksi on pyrittävä heterogeeniseen haastateltavien joukkoon. Erilaista näkemyksen saamiseksi, tutkimuksessa on haastateltu myös yhtä kiinteistön omistajaa. Tutkimukseen osallistujien valintaan käytettiin niin sanottua lumipallomenetelmää, eli haastateltavat kysyttiin mukaan tutkimukseen, jonka jälkeen heiltä kysyttiin mielipidettä sopivasta haastateltavasta. Näin pystytään hyödyntämään ammattilaisten kontakteja löytämään päteviä haastateltavia. [22, p. 18–19]

Haasteluteemat olivat

- mitä erityisen hyvää järjestelmässä on
- mikä on järjestelmän pääasiallinen käyttötarkoitus
- mitä ongelmia järjestelmässä on
- millainen käyttöliittymä järjestelmässä on
- mitä tulevaisuuden suunnitelmia on joko tälle kiinteistöautomaatiojärjestelmälle tai tuleville järjestelmille?

Teemahaastatteluissa käytetty kysymysrunko on tämän työn liite 1B. Jokaiseen kysymykseen ei kaikilta haastateltavilta saatu vastausta, koska haastateltavien asema omassa organisaatiossaan oli vaihteleva. Osaltaan tämä kertoo siitä, että kiinteistöautomaatiosta ei välttämättä ole riittävästi tietoutta kiinteistöjen viiteryhmissä.

Tutkimuksen ensimmäinen kohde oli kunnan kiinteistöautomaatiojärjestelmäkeskuksen valvomotyöntekijän haastattelu. Kyseissä etävalvomossa on eri-ikäisiä järjestelmiä kuudelta eri kiinteistöautomaatiojärjestelmän toimittajalta. Hallinta tapahtuu Internet-yhteyttä hyödyntäen. Valvottavia kohteita ovat muun muassa koulut, asuinrakennukset ja liikuntahallit. Hallittavia kohteita on kymmeniä. Eri toimittajien järjestelmät toteuttavat samat asiat, jotka on sovittu jo tilausvaiheessa. Myöskään laatueroja eri toimittajien välillä ei ole juuri koettu. Käytännössä kunnan järjestelmillä on ainoastaan kaksi hankintakriteeriä: järjestelmän pitää täyttää ennalta sovitut määritelmät, jonka

jälkeen valitaan halvin toimittaja tarjouskilpailun jälkeen. valvomotyöntekijän mukaan kiinteistöautomaatio on todella kilpailtu ala. Haastattelun valvomotyöntekijän mukaan, poliisikoulun rakentamisen kokonaisbudjetista 3 % prosenttia meni kopiointikuluihin ja 1 % kiinteistöautomaatiojärjestelmään. Pääasiallinen tarkoitus kiinteistöautomaatiolla on energiansäästäminen ja siten taloudellisuus. [Liite 1B, a] Energiansäästäminen ja kustannustehokkuus oli myös kaikkien muiden haastateltavien mukaan järjestelmän päätarkoitus. [Liite 1B, b, c, d, e, f]

Toinen haastattelukohde oli vuokra-asuntosäätiön kiinteistöautomaatiovalvomon valvomotyöntekijät. Haastattelun yhteydessä käytiin tutustumassa kahteen kerrostaloon: uuteen ja vanhaan. Etävalvomossa on järjestelmiä kuudelta eri toimittajalta. valvomotyöntekijöiden mukaan, pyrkimyksenä on kuitenkin vähentää eri järjestelmiä muutamalla. Ei kuitenkaan ainoastaan yhteen järjestelmään, jotta toimittajien kesken säilyy kilpailutilanne. Erityistä huomiota yrityksessä pyritään kiinnittämään urakoitsijoiden valintaan – useimmiten hyödynnetään tuttuja ja luotettavaksi havaittuja urakoitsijoita. [Liite 1B, b]

Koulurakennuksen virkaa haastattelussa toimitti yliopistorakennus. Haasteltavana henkilönä oli kiinteistön huoltomies. Kampuksella on monen ikäisiä rakennuksia, jotka pyritään yhdistämään samaan valvomoon. Huoltomiehen kuvailujen perusteella, varsinkin uudemmissa rakennuksissa on myös erikoisempia hallittavia kohteita. Vanhoissa taas tekniikaltaan hieman vanhentuneita laitteita, jotka joskus aiheuttavat ongelmia. [Liite 1B, c]

Seuraava kohde oli juhlatalo ja kaksi kerrostaloa. Haasteltavana oli kiinteistönhuoltomies. Tämän huoltomiehen mukaan, kiinteistöautomaatiojärjestelmän tärkeimmät tehtävät kohteessa olivat rakennuksen valvominen ja tarkkailu: vikatilanteiden löytäminen ja niistä hälyttäminen. Huoltomies kuvasi järjestelmiä huoltomiehen työkaluksi, joka ei itsessään ole riittävä kiinteistön hallintaan. [Liite 1B, d]

Tehdashallikohteena toimi yritys, joka tarvitsee tuotteidensa valmistukseen suuren hallitilan ja siihen yhdistettyjä toimistotiloja. Yhteensä kohteessa on kaksi tehdashallia ja yksi toimistorakennus. Rakennukset on rakennettu eri aikaan ja niissä on erillinen automaatiojärjestelmä. Meneillään olevana projektina on järjestelmien yhdistäminen. Haasteltavana oli kiinteistönhallintapäällikkö, jolla ei ollut suoraa kosketusta automaatiojärjestelmän hallintaan. Energiansäästämisen lisäksi järjestelmän tärkeimpänä ominaisuutena, kiinteistöpäällikön mukaan, on olosuhteiden hallinta. [Liite 1B, e]

Viimeisenä kohteena oli pienehkö ravintola, jossa on oma kiinteistöautomaatiojärjestelmä. Haasteltavana henkilönä oli kiinteistönomistaja. Omistaja ei ollut vielä opetellut käyttämään vähän aikaa sitten asennettua järjestelmää, mutta piti sen olemassa oloa kuitenkin tarpeellisena olosuhteiden hallinnan vuoksi. [Liite 1B, f]

Haastattelututkimuksen kohteiden luonteesta saadut tulokset vahvistivat aieman esitutkimuksen käsitystä siitä, mikä kiinteistöautomaatiojärjestelmissä on tärkeää. Lähes kaikissa kohteissa energianhallinta ja sen avulla luotavat säästöt olivat tärkeimmässä asemassa, kuitenkin unohtamatta olosuhteiden hallintaa. Tulos pysyi samana, riippumatta kohteen luonteesta ja koosta sekä haastateltavan henkilön roolista. Huomi-

oitavaa on kuitenkin, ettei mikään järjestelmä onnistunut erityisesti erottautumaan edukseen tässä yhtenäisessä joukossa. Myös markkinoiden kovasta kilpailusta tuli hyviä esimerkkejä. Monessa kohteessa oli järjestelmiä usealta toimittajalta. Selvänä trendinä on päivittää ja yhdistää vanhentuneet järjestelmät osaksi uutta järjestelmää. Erityisen kuvaavaa oli se mitä kunnalla töissä oleva valvomotyöntekijä kertoi kiinteistöautomaatiojärjestelmille varatusta budjetista. Kuitenkin kiinteistöautomaatiojärjestelmät koetaan oleelliseksi osaksi rakennusta.

4.3.1 Järjestelmien positiiviset ja negatiiviset puolet

Tutkimuksen kohteena olleiden kiinteistöjen kiinteistöautomaatiojärjestelmien positiiviset puolet jakaantuvat sekä yleisen tason huomioihin sekä yksityiskohtaisempiin huomioihin. Kaikkien viiteryhmiä mielestä oli hyvä, että kiinteistöautomaatiojärjestelmä on keskittynyt tiettyyn asiaan kuten energianhallintaan. Myös järjestelmään jätetty laajenusvara tulevaisuutta koettiin erityisen positiiviseksi. [Liite 1B] Yksityiskohtaisemmat huomiot tulivat erityisesti järjestelmien valvojien käyttäjiltä. He muun muassa arvostivat ominaisuuksia, jotka tarjoavat järjestelmästä mahdollisimman paljon tietoa. Esimerkiksi mitä enemmän mittauskäyriä kiinteistötoiminnasta oli mahdollista piirtää, sitä parempi. Valvomoiden käyttäjät painottivat erityisesti myös käyttöliittymän selkeää suunnittelua: mitä loogisemmin kasattuja ja yleisiä käytäntöjä noudattavia ne ovat, sitä parempi. Pääsyy on se, että monessa valvomossa on usean eri toimittajan järjestelmiä. Kun ne pääpiirteittäin muistuttavat toisiaan, niiden käyttäminen on helpompaa. Käyttöliittymässä mielipiteitä jakava kysymys oli piirrosmerkkien vai graafisempien kuvien käyttäminen käyttöliittymän kaavioissa. Piirrosmerkkejä kuvattiin selkeämmiksi osittain siitä syystä, että ne noudattavat alan standardeja. Henkilölle, jolla ei ole paljon kokemusta järjestelmän käyttämisestä, graafisempi kuva voi olla helpompi. Toinen tärkeä huomio on mahdollisuus etävalvomon käyttämiseen eli web-käyttöliittymän hyödyntäminen [Liite 1B]

Yleisellä tasolla suurimmat kiinteistöautomaatiojärjestelmiin liittyvät ongelmat, tutkimuksen mukaan liittyivät rakennusprojektin organisointiin. Suomessa kaapelointi kuuluu sähköurakkaan ja kiinteistöautomaatio automaatiourakkaan. Koska kaapelointi on erillään automaatiourakasta, on anturoinnin muuttaminen hankalaa. Erään valvomotyöntekijän mukaan Ruotsissa sähkö- ja automaatiourakat ovat yhdistetty ja suunnittelu on jouhevampaa. Toinen yleinen organisointiin liittyvä ongelma kiinteistöautomaation suunnittelun tekemisen yrityksen ja urakoitsijan välisen yhteistyön toimimattomuus: suunnitelma ei vastaa aina niitä vaatimuksia, jotka urakoitsija on velvoitettu tekemään. Standardoinnin ongelmat tulivat esiin erityisesti korjausrakentamisessa – vanhojen järjestelmien rajapintojen yhdistämisessä uudempaan järjestelmään oli ongelmallista. [Liite 1B, a, b]

Kiinteistöhuoltajien keskuudessa suurimmat ongelmat olivat riittävän koulutuksen, motivoinnin ja luottamuksen puute. Lisäksi huoltomiehet kokivat, että automaatio on usein myöhässä. Kun ohjelma hälyttää, vahinko on usein jo tapahtunut. He kaipaavat enemmän ennakoivaa automaatiota. Kiinteistöhuoltajat kokivat myös jotkut ominai-

suuksista vaikeakäyttöisiksi. Esimerkiksi aikatauluihin perustuvan ohjauksen käyttöliittymän hallinnassa oli usein vaikeuksia. Myös eräs valvomotyöntekijä myönsi, että jokaisessa käytettävässä järjestelmässä on enemmän ominaisuuksia kuin mitä käytetään. [Liite 1B, b, c, d]

Tärkeitä huomioita haastattelukohteiden positiivisista puolista on, että kaikki järjestelmät oli koottu loogisesti vastaamaan alan yleisiä käytänteitä. Tämä auttaa vieraiden järjestelmien omaksumista, mutta toisaalta vaikeuttaa alan kilpailua. Korostettu vaatimus oli järjestelmän selkeys etenkin käyttöliittymän osalta. Suurimmat ongelmat liittyivät selvästi eri viiteryhmiä yhteistyön puutteeseen. Ongelmaan vastaaminen tarkoittaa muun muassa koulutuksen lisäämistä ja asiakkaan kuuntelua tuotekehitysvaiheessa.

4.3.2 Tulevaisuus haastattelujen perusteella

Haastattelututkimuksen analyysissa on lähdettävä liikkeelle tutkimuskysymys edellä. Tässä haastattelussa tutkimuskysymyksenä oli selvittää millaisia hyviä ja huonoja puolia nykyään käytössä olevissa järjestelmissä on ja mitä niiltä odotetaan tulevaisuudessa. Analysointi tapahtuu yhdistelemällä eri haastateltavien vastauksia: mitä yhteistä ja eroavia asioita haastatteluista löytyi? [22, p. 22]

Haastatteluiden perusteella kiinteistöautomaation tulevaisuus tarkoittaa automatisoitavien kohteiden lisäämistä, lisääntyviä mittauksia ja valvontaa sekä etäkäytön yhä parempaa huomioimista. Automaation lisääminen tulevaisuudessa oli selkeä trendi kaikkien haastateltavien mielestä, paitsi haastatellun kiinteistön omistajan, jolla ei ollut juurikaan tietoutta kiinteistöautomaatiosta. Automatisoinnin lisääminen tulee haastattelujen perusteella näkymään erityisesti parempana energian hallintana ja lisääntyvänä tietoutena ohjattavasta kohteesta. Sekä valvomotyöntekijöiden että huoltomiesten selkeä toive oli anturoinnin lisääntyminen ja siten informaation kerääminen järjestelmästä. Tämä tulee tarkoittamaan myös yhä parempaa dokumentointia. [Liite 1B]

Myös kehittyneemmän ja nopeamman laitteiston käyttäminen oli selkeä tavoite kohteissa. Käytännössä tämä tarkoittaa käyttöliittymän sujuvampaa toimintaa ja kehittyneempiä kosketusnäyttöpaneeleja. Toinen yleinen laitteiston kehittymiseen liittyvä trendi oli langattoman anturoinnin lisääminen kiinteistöissä. Tällä hetkellä missään kohteessa ei vielä ollut langattomia antureita. Eräässä kohteessa oli ollut kokeiluja, jotka eivät vastanneet odotuksia, joten langattomista antureista luovuttiin. Langattomien antureiden mahdollisuudet nähtiin kuitenkin niin suuriksi – etenkin korjausrakentamisessa – että niiden käyttöönotto kiinteistöissä tulee tapahtumaan tulevaisuudessa. Ne tuovat potentiaalisesti paljon joustavuutta kiinteistöautomaatiojärjestelmien hallintaan. Toinen selkeä suuntaus oli etäkäytön lisääntyminen ja parempi mahdollistaminen. Eräs haastateltava painotti kuitenkin tietoturvan tärkeyttä myös kiinteistöautomaatiojärjestelmissä ja halusi, että työn tässä osiossa nostetaan esiin Aalto yliopiston Seppo Tiilikaisen ja Jukka Mannerin raportti: *Suomen automaatioverkkojen haavoittuvuus – Raportti julkisesti esillä olevista automaatiolaitteista* (2013), mistä voi lukea lisää asiasta. [Liite 1B, a, b]

Haastattelututkimuksen tulevaisuuden visiot osuvat hyvin ennakkokäsityksiin, jotka oli saatu kirjallisuustutkimuksen perusteella. Kiinteistöt ovat kehittymässä kohti älytaloa – niistä tahdotaan saada yhä enemmän informaatiota käytettäväksi. Myös haastattelututkimuksen tulokset ovat rohkaisevia yrityksen oman kiinteistöautomaatiojärjestelmän suunnittelun kannalta.

4.4 Kiinteistöautomaatiojärjestelmän tavoitteet

Alkuperäinen tuotteistusidea kiinteistöautomaatiojärjestelmästä on käynyt läpi sekä tausta- että markkinatutkimuksen. Vaiheisiin on sisältynyt sekä kirjallisuustutkimuksen suorittamista että laadullista haastattelutyötä. Selvityksestä saatujen tietojen perusteella on järjestelmällä voidaan asettaa tavoitteet.

Tärkeimpänä tavoitteena on luoda järjestelmä, joka toimii mahdollisimman monenlaisessa kohteessa – näin mahdollistetaan suuri asiakaskunta. Vahvasti kilpailulla markkinalla on etua, jos on mahdollista muuntautua ja joustaa. Mahdollisia automatisoitavia projekteja tulevaisuudessa ovat kerrostalot, toimistot, sairaalat, kauppakeskukset ja muut riittävän suuret rakennukset. Onnistuneista suurien kohteiden automatisointiprojekteista on mahdollista saada hyviä referenssejä, joita voidaan hyödyntää markkinoinnissa. Kyseessä on siis skaalautuva järjestelmä, jota ei rajoita kohteen koko tai automatisoitavat toiminnot. Koska asiakkaiden vaatimukset ja saatavilla olevat toimilaitteen ovat hyvin erilaisia, on järjestelmän tuettava mahdollisimman useaa erilaista kenttäväylää ja liitää sekä oltava laajennettavissa myöhemmin.

Kotiautomaatioon ei tässä järjestelmässä keskitytä. Kiinteistöautomaation hinta on alhainen, joten pienistä kohteista ei saada riittävästi tuottoa. Lisäksi kuten esimerkiksi Marttilan kyselytutkimuksessa todetaan, kotiautomaation suosiossa ja yleistymisessä on vielä haasteita.

Yleisellä tasolla järjestelmältä vaaditaan vähintään alalla vakiintuneita toiminnallisuuksia, esimerkiksi lämmönsiirtojärjestelmän, ilmastoinnin ja valojen hallinnan automatisointia. Järjestelmän täytyy olla riittävän tarkka, toimintavarma ja pitkäikäinen. Vain oikein toimivalla järjestelmällä saadaan hyvä maine asiakkaiden keskuudessa. Jo nyt suurinta osaa järjestelmistä on mahdollista ohjata etäohjauksella Internetin kautta. Kehityssuunta on laajenemassa koskettamaan yhä monipuolisemmin erilaisia mobiililaitteita. Kehitetyn järjestelmän on siis pystyttävä vastaamaan alan yleiseen suuntaukseen.

On kuitenkin muistettava, että pelkillä perusominaisuuksilla ei ole järkevää lähteä mukaan kilpailulle alalle. Uudet direktiivit vaativat uusien rakennusten olevan yhä energiatehokkaampia. Energianhallinta on myös rakennusten omistajia kiinnostava ominaisuus sen potentiaalisesti tuomien säästöjen vuoksi. Yrityksessä on myös sisäisesti tietoutta ja osaamista energianhallinnan vaatimuksista ja toteutustavoista. Näiden asioiden vuoksi on tavoitteena kehittää järjestelmään erikoisominaisuuksia, jotka liittyvät kiinteistön energianhallintaan, -raportointiin ja -mittaamiseen.

Varsinkin haastattelututkimuksessa nousi esiin toive helppokäyttöisestä järjestelmästä. Kyseinen vaatimus koskettaa erityisesti käyttöliittymää, sillä se on osuus, jonka kanssa loppukäyttäjä tulee kommunikoidaan. Säädot, järjestelmän tilatiedot, muut toiminnot ja järjestelmän tarjoama informaatio on oltava loogisesti ja helposti löydettävissä. Helppokäyttöisyyttä ei voi yksilöidä tiettyyn toiminnolliseen rakenteeseen, vaan se on osa tuotteen kokonaisuutta.

Edelliset tuotteelle esitetyt vaatimukset koskettavat asiakasrajapintaa eli niitä asioita, joita tuotteen on tarkoitus tarjota asiakkaalle. Jotta itse tuotteen suunnittelu, jatkokehittäminen ja tuotanto olisi mahdollisimman sujuvaa, helppokäyttöisyyden täytyy ulottua myös ohjelmistotasolle. Suunnittelun ohjelmiston täytyy olla myös ohjelmoijalle ja suunnittelijalle helppokäyttöinen. Vaatimuksen tärkeyttä korostaa se, että järjestelmää tuskin kehittää tai käyttää vain yksi henkilö. Sen vuoksi tiedon on oltava dokumentoitua ja mahdollisimman rakentavasti esitettyä. Luvussa 2 on käsitelty tuoterunkoarkkitehtuurien etuja verrattuna perinteiseen ohjelmistokehitykseen. Tavoitteena on luoda kiinteistöautomaatiojärjestelmälle tuoterunkoarkkitehtuuri, jonka avulla on mahdollista toteuttaa projekteja nopeasti ja kustannustehokkaasti. Samalla tuoterunkoarkkitehtuurien mahdollisuudet ja käytännöt tulevat esille myös muiden alojen projekteja varten.

Yrityksen itsensä kannalta tavoitteena on luonnollisesti päästä kiinteistöautomaatiomarkkinoille mukaan tuoteportfolioa laajentamalla. Tausta-ajatuksena on myös yrityksen sisäisen osaamisen parempi hyödyntäminen ja yhteistyön lisääminen yrityksen asiantuntijoiden välillä.

4.5 Tuotekehityksen organisointi ja komponentit

Kun on suunniteltu tarvittavat järjestelmälle asetetut vaatimukset, on syytä pohtia miten järjestelmä toteutetaan fyysisesti: millaisia ja kenen valmistajan komponentteja käytetään ja millaisia ohjelmistoja toteutuksessa käytetään? Kysymyksiä pohditaan tutkimalla seuraavia kysymyksiä:

- komponenttien tehokkuusvaatimuksia
- mahdollisuudet sujuvan käyttöliittymän rakentamiseen
- erilaisille väyläliitännöille, toimilaitteille tai muille vastaaville, tarjolla olevaa tukea
- kuinka monta automaatiopisteitä järjestelmään on mahdollista saada
- yrityksen sisäisiä valmiuksia omaksua komponenttien ja ohjelmistojen käyttö.

Komponenttien tehokkuusvaatimus on hyvin yksinkertainen. Komponenttien täytyy olla niin suorituskykyisiä, että järjestelmän toiminta on sujuvaa. Ohjelman täytyy toimia komponenteilla sujuvasti ja erityisesti käyttöliittymän toiminnan täytyy olla jouheaa. Kiinteistöautomaatiojärjestelmä toteutetaan PLC-pohjaisena, koska yrityksellä on PLC:iden käyttämisestä paljon kokemusta. PLC on hyvin perinteinen tapa toteuttaa automaatiojärjestelmiä – myös kiinteistöautomaatiota. Komponenttien tehokkuusvaatimus koskee erityisesti PLC:tä, sillä se on koko järjestelmän perusta. Sujuvan käyttöliittymän komponenttivaatimukset liittyvät erityisesti valittavaan näyttöpäätelaitteeseen –

toki myös käyttöliittymän sujuvuuteen liittyy myös PLC:n suorituskyky. Näytön on oltava suuri kosketusnäyttö. Noin 10 tuuman kokoinen näyttö on haastattelututkimuksen mukaan sopivan kokoinen näyttö järjestelmälle. Myös näytön komponenttien on oltava riittävän suorituskykyinen, jotta käyttöliittymäohjelman toimiminen on sujuvaa. Vaatimus mahdollisimman laajasta tuesta erilaisiin väyläliitäntöihin pyrkii vastaamaan tutkimuksessa esiin tulleeseen ongelmaan kiinteistöautomaatiojärjestelmien alhaisesta standardoinnin tasosta. Markkinoilla on hyvin paljon käyttökelpoisia kenttäväyliä ja niitä tukevia toimilaitteita. Mitä laajempi tuki eri liitännöille saadaan jo komponenttitasolla sitä helpompaa on ohjelmallinen toteutus. Automaatiopisteiden lukumäärävaatimus viittaa siihen, kuinka monta tuloa ja lähtöä saadaan PLC:hen liitettyä. Järjestelmän tarkoituksena on pystyä toteuttamaan hyvin monenlaisia ja kokoisia järjestelmiä, joten mitä joustavampi komponenttien automaatiopistevaatimus sitä parempi. Yrityksen sisäiset valmiudet omaksua järjestelmän käyttäminen tarkoittaa erityisesti sitä, onko yrityksellä sisäistä osaamista jonkin järjestelmän käyttämisestä erilaisissa projekteissa. Lisäksi on syytä tutkia, onko yrityksellä mahdollisesti valmiita kontakteja johonkin komponenttivalmistajaan. Komponenttivalmistajalta on mahdollista saada tuotteita halvemmallalla sekä parempaa tukea tuotteiden kehitykseen.

Laitevalmistajia on markkinoilla hyvin paljon, eikä niiden kaikkien esittely tässä työssä ole kannattavaa. Käytännössä järjestelmä olisi mahdollista toteuttaa hyvin monen eri komponenttivalmistajan tuotteilla – varsinkin prosessiteho ja näyttöpäätevaatimukset on mahdollista saavuttaa usealla eri komponentilla. Eroja kuitenkin syntyy siinä, miten hyvin komponentit tukevat eri kenttäväyliä ja miten paljon automaatiopisteitä niihin on mahdollista saada kiinni. Jesse Ylitalon insinöörityössä *Rakennusautomaation väylät integraatio* (Metropolia, 2010), on tehty kuudelle eri laitevalmistajalle ja urakoitsijalle kysely siitä, mihin järjestelmä pystyy. Ylitalon kyselystä nousee esiin, että Beckhoff Automation Oy:n ja Siemens Oy:n tuotteet, jotka tukevat kaikista laajimmin kenttäväyliä ja laajentumismahdollisuuksia.

Molempien laitevalmistajien tuotteilla olisi mahdollista toteuttaa järjestelmä. Tämän työn toteuttavaksi laitteistoksi valittiin kuitenkin Beckhoffin järjestelmä, koska yrityksellä on kokemusta kyseisen valmistajan komponenttien käyttämisestä. Toinen erittäin painava syy on se, että yrityksellä ja Beckhoffilla on kumppanuussopimus, jonka ansiosta on mahdollista saada koulutusta ja tuotetukea järjestelmän rakentamiseen. Beckhoffilla on varsinkin Saksassa laaja kokemus rakennusautomaatioprojekteista. Järjestelmä ei periaatteessa aseta rajoja sille, mitä on mahdollista automatisoida. [33]

Komponenttivalmistajan valinnan jälkeen on pohdittava, millä ohjelmistolla tuotekehitysprojekti toteutetaan. Valinta luonnollisesti rajoittuu valitun laitevalmistajan tukemiin ohjelmistoihin. PLC:n ohjelma on mahdollista toteuttaa TwinCat 2- tai TwinCat 3-ohjelmistoilla. Molemmat ohjelmistot tukevat PLC-ohjelmoinnin standardia IEC-61131:stä, eli käytettävissä on useita eri ohjelmointikieliä. TwinCat 3 on kehittyneempi ja uudempi versio TwinCat 2-ohjelmistosta, joten se vaikuttaa luonnolliselta valinnalta. Esimerkiksi käyttöliittymään on tehty paljon parannuksia ja ohjelma tukee myös IEC-61131:n olio-ohjelmointipiirteitä, joita voisi varmasti hyödyntää kiinteistöautomaation

ohjelmoinnissa. Kun valintaan ajatellaan tarkemmin kiinteistöautomaation kannalta, nousee kuitenkin esiin yksi erittäin tärkeä seikka: TwinCat 2 -ohjelmistoon on Beckhoffilla olemassa kiinteistöautomaatiofunktiokirjasto. Kyseistä kirjastoa ei ole vielä käännetty TwinCatin uudemmalle versiolle. Siinä on valmiita ratkaisuja esimerkiksi pumpujen ja venttiilien ohjaukseen ja paljon yleisiä kiinteistöautomaatiojärjestelmien tarvitsemia toimintoja. Kirjaston avulla saadaan ohjelmiston kehitystä nopeutettua huomattavasti. Vaikka komponentteja ei välttämättä voikaan suoraan hyödyntää ohjelmiston toteutuksessa, jokaisen komponentin ohjausta ei kuitenkaan kirjastoa hyödyntämällä tarvitse kehittää alusta asti. [33]

TwinCat 2 -ohjelmistolla olisi mahdollista luoda myös järjestelmän käyttöliittymä. Käyttöliittymää ei ole kuitenkaan mahdollista saada TwinCatilla riittävän modernin näköiseksi. Tämän vuoksi käyttöliittymän toteuttavaksi ohjelmaksi valitaan Beckhoffin suosituksesta Wonderwaren Indusoft -ohjelmisto. Kyseisellä ohjelmalla on mahdollista toteuttaa hyvin monenlaisia käyttöliittymätoteutuksia. Erityisen tärkeää on se, että ohjelmassa on sisäänrakennettu tuki Beckhoffin PLC:iden kanssa keskusteluun, joten kyseistä ominaisuutta ei tarvitse rakentaa itsenäisesti. Indusoft tukee myös Internet- ja mobiilikäyttöliittymätoteutuksia. Ohjelma kuuluu myös Beckhoffin tuen piiriin, joten sen myös sen käyttämiseen on mahdollista saada apua. [33]

Käytettävät ohjelmistot takaavat sen, että järjestelmän toteuttaminen on mahdollista ja toteutukseen on saatavilla osaamista sekä yrityksen sisältä että ulkopuolisilta asiantuntijoilta. Osaaminen ei ole kuitenkaan itsestään riittävä tausta tuotekehityksen onnistumiselle. Edelleen tuotekehitykseen liittyy riski epäonnistumisesta. Riskiä pyritään madaltamaan jatkamalla innovointiprosessin hyödyntämistä myös vaativaan tuotekehitysvaiheeseen. Tarkoituksena on tuotekehityksen jatkuva arviointi, tiedon jakaantuminen organisaatiossa ja positiivisen ilmapiirin aikaansaaminen. Tuoterunkoarkkitehtuurin kehitysprosessi jaksetaan samalla tavalla kuin koko tuotekehitys- ja ideointiprosessi. Tuoterunkoarkkitehtuurin kehittämiseksi asetetaan ajallisia ja sisällöllisiä tavoitteita, joiden saavuttamista tutkitaan palaverissa ja kokouksissa. Palaverissa tuloksia verrataan ennalta määritellyyn aikatauluun ja tavoitteisiin. Näin koko tuotekehitysprosessi pysyy paremmin hallinnassa.

5 KIINTEISTÖAUTOMAATIOJÄRJESTELMÄN TUOTERUNKOARKKITEHTUURIN KEHITYS

Markkinatutkimuksen jälkeen siirrytään projektin aikavievimpään ja haastavimpaan vaiheeseen: ohjelmistonkehitykseen, eli innovointiprosessin vaiheeseen 3. Vaiheen tarkoituksena on tuottaa mahdollisimman valmis ja testattu tuoterunkoarkkitehtuuri kiinteistöautomaatiojärjestelmälle. Kuten luvussa 2 todettiin, innovointiprosessin vaativimmat vaiheet on syytä jakaa pienempiin kokonaisuuksiin, jotta projektin toteuttaminen olisi mahdollisimman jäsennettyä, kontrolloitua ja seurattavissa. Tuotekehitysvaiheen sisävaiheet ovat luvussa 3 esitetyt tuoterunkoarkkitehtuurin kehitysvaiheet: esitutkimus, vaatimusten määrittely sekä alustankehitysprosessi. Kun nämä vaiheet on suoritettu, palataan takaisin koko projektin innovointiprosessimalliin. [13, p. 164]

5.1 Tuoterunkoarkkitehtuurin esitutkimus

Tuoterunkoarkkitehtuurin kehityksen ensimmäinen vaihe on esitutkimus, joka on hyvin samanlainen vaihe kuin koko projektin markkina- ja esitutkimusvaihe. Erona on, että se koskettaa ainoastaan tuoterunkoarkkitehtuurin ohjelmiston kehitystä. Pääpaino on selvittää, millaisia toiminnallisuuksia ja vaatimuksia tuoterunkoarkkitehtuurilta vaaditaan. Lisäksi on pohdittava tuoterunkoarkkitehtuurin hyödyt ja haitat: onko tuoterunkoarkkitehtuuria järkevä toteuttaa kiinteistöautomaatio tuoteperheelle? On siis pyrittävä pohtimaan myös tulevaisuuden tarpeita. Lisäksi esitutkimusvaiheessa on mietittävä organisaation tason suunnittelu- ja toteutusprosessi. Käytännössä on järjestettävä projektin etenemisen seuranta ja laadittava välitavoitteet. Aiheista suoritettavien pohdintojen taustalla on neljän eri lähteen tarkastelu: yrityksen sisäinen osaaminen, Beckhoffin rakennusautomaatioasiantuntijat, suoritettu markkinatutkimus ja muu esitutkimus.

Tuoterunkoarkkitehtuurin tarpeellisuuden pohdinta on esitutkimuksen ensimmäinen vaihe, koska se määrittää, toteutetaanko järjestelmää alkuunkaan suunnitellulla tavalla. Tähän astisessa tutkimuksessa on selvinnyt, että kiinteistöautomaatiojärjestelmillä on useita toistuvia rakenteita: käytännössä kaikissa järjestelmissä on ilmanvaihtokoneita, lämmönsiirtojärjestelmiä, valojenohjauksia, kalenteriohjauksia ja muita toimintoja. Toistuvat rakenteet ovat tuoterunkoarkkitehtuurin toteutuksen elinehto. Lisäksi yrityksellä on valmiuksia aloittaa tuotekehitysprosessi

Ongelmia kuitenkin syntyy, kun yleisesti toistuvia ominaisuuksia tarkastellaan lähemmin. Jokainen ominaisuus voidaan toteuttaa usealla eri tavalla – eri asiakkailta voi olla myös vaatimuksia erilaisista toteutustavoista. Esimerkiksi suoritetaanko ilmanvaihtokoneen puhaltimien ohjaus taajuusmuuttajilla vai jollain muulla tavalla. Erilaiset vaa-

timukset hankaloittavat tuoterunkoarkkitehtuurin luomista, sillä jokaiseen variaatioon ei ole mahdollista etukäteen varautua. Käytännössä myöskään mikään rakennuskohde ei ole identtinen toisen kohteen kanssa, vaikka ne saattaisivat ulkoisesti siltä vaikuttaa. Tuoteperheen sisällä on siis varianssia.

Eroavaisuudet vaikuttavat tuoterunkoarkkitehtuurin laajuuteen, eli siihen mitä ominaisuuksia tuoterunkoarkkitehtuuriin voidaan kiinteästi liittää. Jos kaikkiin eroavaisuuksiin varauduttaisiin tuoterunkoarkkitehtuurissa, paisuisi se valtavan laajaksi ja siten hankalasti käytettäväksi ja ylläpidettäväksi. On löydettävä tasapaino tuoterunkoarkkitehtuurin laajuuden ja sen sisältämien ominaisuuksien välille. Oletuksena on myös, että tuoterunkoarkkitehtuuri nopeuttaa projektien tekemistä, mutta itse arkkitehtuurin tekeminen on hidasta ja hankalaa. Arkkitehtuurin edut eivät kuitenkaan rajoitu pelkästään ohjelmointitasolle, vaan sen avulla saadaan koko prosessista standardoidumpi: voidaan määritellä miten erilaiset työvaiheet toteutetaan, ja asiakasprojektien hallinnointi on tältä pohjalta helpompaa.

Vaikka toteutukseen liittyy riskejä, ennakkotietojen perusteella kiinteistöautomaatiojärjestelmälle kannattaa toteuttaa tuoterunkoarkkitehtuuri. Tuoterunkoarkkitehtuurin potentiaalisesti tuomat edut ovat suuremmat kuin hallittu riski. Koska kiinteistöautomaatioprojektit ovat yritykselle uusi aluevaltaus, tuoterunkoarkkitehtuuri on hyvä keino helpottaa asiakasprojektien aloittamista – ensimmäistä asiakasprojektia ei tarvitse aloittaa täysin tyhjästä, vaan valmiina on arkkitehtuuripohja. Näin ollen arkkitehtuuri nopeuttaa ja selkeyttää asiakasprojekteja ja samalla tuottaa arvokasta tietoa, miten tuoterunkoarkkitehtuuria on mahdollista jatkokehittää. Asiakasprojektia ei ole tällä hetkellä näkyvissä, mikä on sekä hyvä että huono asia. Koska asiakasprojektia ei ole, vuorovaikutus asiakkaiden kanssa puuttuu. Tätä puutetta korjaa luvussa 4 suoritettu taustatutkimus sekä saatavilla oleva asiantuntijatuki. Positiivinen asia asiakkaan puuttumisessa on, ettei työllä ole kiire – asiakkaan luoma aikataulupaine ei pakota kiirehtimään työn valmistumista.

Esiin nousee kaksi selkeää ongelmaa: tuoteperheen tuotteiden erot toisiinsa nähdessä sekä arkkitehtuurin suunnitteluun kuluva aika. Tuoteperheen tuotteiden erilaisuuteen voidaan vastata tarkalla suunnittelulla. Suunnitteluun kuluva aika taas on yksinkertaisesti hyväksyttävä.

Esitutkimusvaiheeseen kuuluu myös arkkitehtuurin dokumentoinnin ja ohjelmoinnin laatuvaatimusten määrittely. Tuoterunkoarkkitehtuuri tulee jakautumaan kahden toistensa vuorovaikutuksessa olevaan osaan, koska PLC:n ja käyttöliittymän ohjelmat toteutetaan eri ohjelmistoilla. Tämä toteutustavan vuoksi alustankehitysprosessi on jaettu kahteen vaiheeseen: PLC:n ja käyttöliittymän tuoterunkoarkkitehtuurin suunnitteluun. Nämä vaiheet taas on jaettu pienempiin kokonaisuuksiin, joilla pyritään selkeyttämään tuoterunkoarkkitehtuurin kehittämistä ja helpottamaan sen hallinnointia.

5.2 Tuoterunkoarkkitehtuurin vaatimukset

Tuoterunkoarkkitehtuurin laajuuden määrittää pääasiassa tuettavat ominaisuudet. Tarkoituksena on luoda kompromissi tuoterunkoarkkitehtuurin sisältämien ominaisuuksien ja ohjelmiston koon välillä. Vaatimukset määritellään pääasiassa markkinatutkimuksessa tapahtuneen selvityksen avulla, mutta myös muu esitutkimusaineisto, asiantuntijoiden mielipiteet ja yrityksen oma sisäinen osaaminen otetaan huomioon.

Tärkeä kysymys on, ovatko edellä mainittujen lähteiden painoarvot yhtä suuret. Kirjallisuudesta saadaan hyvä yleiskäsitys kiinteistöautomaation toiminnasta, mutta yksityiskohtaisemmat asiat kuten ohjelmointityö jää kirjallisuudessa abstraktiksi. Kilpailijoiden järjestelmiä tutkiessa taas saadaan hyvä yleiskuvaus kiinteistöautomaatiojärjestelmistä: yleisistä toiminnollisuuksista, tulevaisuuden visioista ja asiakkaiden vaatimuksista. Pintaa syvemmälle, eli toteutustasolle, ei kuitenkaan tässäkään metodissa päästä. Luonnollisesti kilpailevat yritykset tahtovat pitää toteutustapansa salaisuuksina. Mitä lähemmäs järjestelmän toteutustasoa päästään, sitä tärkeämmiksi lähteiksi muodostuvat kiinteistöautomaatioasiantuntijat. He osaavat vastata kysymykseen, miten ominaisuudet kannattaa toteuttaa ja millaisia työkaluja käytetään. Myös yrityksen sisäistä osaamista on mahdollista hyödyntää. Rakennussuunnittelupuolella on jo aiemmin suunniteltu kiinteistöautomaatiojärjestelmien vaatimuksia, joita voidaan hyödyntää oman järjestelmän kehityksessä.

Tuoterunkoarkkitehtuurin vaatimukset voidaan jaotella karkeasti kahteen osaan: perusominaisuuksiin ja erikoisominaisuuksiin. Molemmat näistä vaatimuskokonaisuuksista kuuluvat sekä PLC:n että käyttöliittymän tuoterunkoarkkitehtuureihin. Perusominaisuudet ovat ominaisuuksia, jotka ovat käytännössä jokaisessa kiinteistöautomaatiojärjestelmässä. Ne vastaavat alan yleisiin käytäntöihin ja standardeihin eli ovat toistuvia ominaisuuksia. Toistuvat ominaisuudet taas kannattaa kiinnittää tuoterunkoarkkitehtuuriin. Erikoisominaisuudet ovat ominaisuuksia, jotka ovat usein asiakkaiden erityisvaatimuksia, eli niiden toistumista on vaikea ennustaa. Pelkillä perusominaisuuksilla ei kuitenkaan voida kilpailla: tarvitaan jotain millä erottautua joukosta ja siihen erikoisominaisuudet tarjoavat mahdollisuuksia. Toisaalta markkinoilla ei käytännössä voida saada jalansijaa perusominaisuuksia, jotka ovat vakiintuneet alan käytännöiksi. Taulukossa 5.1 on esitetty kiinteistöautomaation tuoterunkoarkkitehtuurille suunnitellut perusominaisuudet.

Taulukko 5.1: *Kiinteistöautomaatiojärjestelmän tuoterunkoarkkitehtuurin perusominaisuudet luvussa 1 määritetyllä automaatiotasolla.*

Ilmanvaihtokone	Lämmönsiirtojärjestelmä	Sähköjärjestelmät	Ohjaukset
Lämmöntalteenotto	Kesäsulku	Valaistuksenohjaukset	Kalenteriohjaus
Palopellit	Lämpötilansäädöt		Manuaaliohjaus
Lämpötilansäätö			Lämpötilakäyrät
Kanavapainesäätö			
Sarjasäätö			

Käytännössä jokaisessa kiinteistöautomaatiojärjestelmässä on olemassa ilmanvaihtojärjestelmä, jonka tehtävänä on huolehtia rakennuksen sisäilman laadusta. Ilmanvaihtoon liittyy lämmöntalteenotto, jolla pyritään ottamaan poistoilmasta talteen lämpöenergiaa. Palopeltien tarkoitus on estää tulipalojen leviäminen rakennuksessa. Jos tulipalo havaitaan, palopellit sulkeutuvat ja eristävät tulipalon. Ilmanvaihtoon liittyy myös tuloilman lämpötilan ja kanavapaineen hallinta. Sarjasäätö viittaa tuloilman lämpötilan hallintaan. Sarjasäätö voi toimia esimerkiksi kaksivaiheisesti. Ensin lämpötilaa säädetään lämmöntalteenottojärjestelmällä. Vain jos lämmöntalteenoton tuottama lämmitys- tai jäähdytysteho ei riitä, käytetään muita mahdollisia energialähteitä kuten kaukolämpöä.

Lämmönsiirtojärjestelmään kuuluu sen eri osioiden lämpötilan säädöt kuten käyttöveden ja ilmanvaihtoverkoston säädöt. Kesäsulku taas tarkoittaa kaukolämpöveden katkaisua lämpimän vuodenajan ajaksi.

Sähköjärjestelmien ohjauksiin lukeutuvat erilaiset valaistuksen ohjaukset, esimerkiksi ulkovalojen aikaohjelmaan perustuva ohjaus. Koko järjestelmää taas tulee voida ohjata kalenteriin perustuvasti: voidaan määrätä päivät ja kellonajat, jolloin esimerkiksi ilmanvaihtokoneet ovat päällä. Lämpötilakäyrien hyödyntäminen tarkoittaa käyriä, joiden avulla määritellään säätimen asetusarvo suhteessa ennalta valittuun muuttujaan. Esimerkiksi lämmitysveden lämpötilan asetusarvo voidaan määrittää ulkolämpötilan perusteella. Taulukon 5.1 osajärjestelmien tarkemmat toimintaselostukset ovat tämän työn liitteessä 2.

Perusominaisuudet koostuvat toimilaitteiden ohjaamisesta. Eri toimilaitteilla voi olla erilaisia ohjelmointitason tarpeita: esimerkiksi eri lämpötila-anturien mittaustiedot luetaan ja skaalataan eri tavoilla. Samoin pumppuja ja puhaltimia on erilaisia, ja niitä voidaan ohjata eri tavoilla. Asiakkaalla voi olla vaatimuksia esimerkiksi tietyn yrityksen toimilaitteiden käyttämisestä. Tämä tuottaa tuoterunkoarkkitehtuuriin hallitsematonta varianssia. Tämän vuoksi tuoterunkoarkkitehtuurin olisi hyvä tukea eri valmistajien toimilaitteita. Taulukossa 5.2 on esitetty kiinteistöautomaatiojärjestelmän toimilaitteiden perusominaisuudet.

Taulukko 5.2: *Kiinteistöautomaatiojärjestelmän tuoterunkoarkkitehtuurin perusominaisuudet luvussa 1 määritetyllä kenttätasolla.*

Pumput/puhaltimet	Venttiilit	Valaistus	Anturit
Jäätymisenesto Jumiutumisenesto Käynnistyshidastus PID-säätö	PID-säätö	Ajastukset Kirkkauden mukainen säätö	Mahdollisimman laaja tuki eri anturityypeille

Pumppujen ja puhaltimien toimintavarmuuden takaamiseksi tarvitaan varmistavia ominaisuuksia. Tähän liittyy jäätymisenesto, jonka tarkoituksena on vastata tilanteeseen, jossa kaukolämpöverkon veden lämpötila putoaa liian matalaksi. Jumiutumisenestoa

tarvitaan, kun pumppu tai puhallin on ollut kauan käyttämättömänä – tällöin toimilaitte ajetaan hetkeksi käyntiin riippumatta järjestelmän sen hetkisestä tilasta. Käynnistysshidastuksella saadaan pumput ja puhaltimet saavuttamaan maksimikäyttöteho ennalta määritetyssä ajassa. Säädetäpä käytetään perinteisesti PID-säätöä. Venttiileihin ei liity toimilaitetasolla muuta perusominaisuutta kuin PID-säätö. Valaistusta on pystyttävä ohjaamaan ajastusten ja kirkkauden perusteella. Antureille on osoitettu vain yksi vaatimus: tuoterunkoarkkitehtuurin tulisi tarjota mahdollisimman laaja tuki eri antureille.

Taulukoihin 5.1 ja 5.2 on listattu ominaisuudet, joihin pystytään varautumaan ja joiden voidaan olettaa olevan kiinteistöautomaatiojärjestelmässä. Suurimman ongelman kuitenkin tuottavat erikoisominaisuudet, joihin ei pystytä varautumaan. Näin ollen niitä ei kannata lisätä tuoterunkoarkkitehtuuriin. Erikoisominaisuudet on kuitenkin otettava huomioon tuoterunkoarkkitehtuurissa siten, että niiden lisääminen tuoterunkoarkkitehtuurin pohjalta valmistettuun tuotteeseen ei ole hankalaa.

Taulukoissa 5.1 ja 5.2 kuvatut ominaisuudet keskittyvät ennen kaikkea PLC:n tuoterunkoon, koska toimilaitteiden ja automaation osajärjestelmien toiminnot toteutetaan PLC:llä. Tuoterunkoarkkitehtuuriin kuuluu kuitenkin myös PLC:n tuoterunkoarkkitehtuurin kanssa yhdessä toimiva käyttöliittymän tuoterunkoarkkitehtuuri. Taulukossa 5.3 on esitetty käyttöliittymän tuoterunkoarkkitehtuurille asetettavat vaatimukset.

Taulukko 5.3. *Käyttöliittymän tuoterunkoarkkitehtuurin perusominaisuudet.*

Ulkoasu	Suorituskyky	Yleiset
Selkeys Helppokäyttöisyys Ponnahdusikkunat	Sulava käyttö	Käyttäjien hallinta Trendien hallinta Hälytysten hallinta Järjestelmätapahtumat Turvallisustoiminnot

Taulukon ensimmäisen sarakkeen vaatimukset liittyvät käyttöliittymän ulkoasuun. Haastattelututkimuksessa painottui erityisesti vaatimus käyttöliittymän selkeydestä: yhdellä näytöllä ei saa olla liikaa asiaa, tekstin pitää olla riittävän suurta ja käyttöliittymän toiminnan mahdollisimman loogista. Tähän asiaan liittyy myös käyttöliittymän helppokäyttöisyys. Tärkeitä toimintoja ei piiloteta monen valikon alle, vaan ne ovat helposti löydettävissä. Toisaalta selkeys on myös markkinoiden asettama vaatimus: mitä paremmalta tuote näyttää sitä helpompi se on myydä, esimerkiksi tuotteen yleinen värimaailma on suunniteltu silmiä miellyttäväksi. Ponnahdusikkunat liittyvät siihen, miten käyttöliittymän ohjelma rakennetaan. Käyttöliittymän tuoterunkoon tulee varmasti samoista komponenteista useita eri yksilöitä. Ylläpidon kannalta ei ole kestävä, että jokaiselle toimilaitteelle luodaan oma sivu, esimerkiksi puhaltimen käyntitiedon tarkasteluun. Toimilaitetyypille luodaan käyttöliittymäsovellukseen yksi sivu, jota kaikki samanlaiset toimilaitetyypit hyödyntävät. [Liite 1B, a, b, c, d]

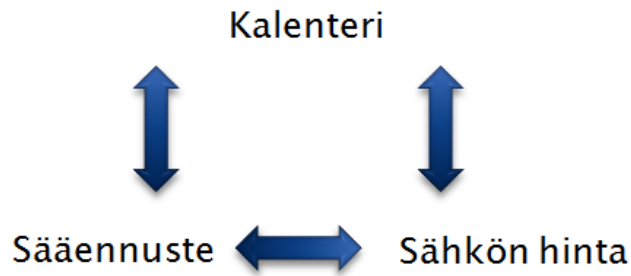
Sarakkeen suorituskykyvaatimus liittyy siihen, miten käyttöliittymän tulee toimia. Sivujen täytyy aueta ripeästi ja viiveettä. Vaatimus ei siis liity ohjelmalliseen toteu-

tukseen vaan komponenttitasoon. Käyttöliittymää ajavan laitteen on oltava riittävän suorituskyykyinen.

Taulukon kolmannessa sarakkeessa on lueteltuna millaisia yleisiä toiminnallisia vaatimuksia käyttöliittymällä on. Käyttäjienhallinta tarkoittaa toimintoja, joilla järjestelmälle voidaan luoda uusia käyttäjiä, poistaa vanhoja käyttäjiä, tarkastella käyttäjien tekemiä toimintoja ja niin edelleen. Trendien hallinta taas tarkoittaa, että käyttöliittymässä on osio, jossa valvomotyöntekijä pystyy tarkastelemaan tiettyjen toimintojen, esimerkiksi pumpun toimintaa ajan suhteen, vertaamaan toimintaa muihin toimilaitteisiin ja niin edelleen. Haastattelututkimuksen mukaan tämä on todennäköisesti valvomotyöntekijöiden tärkein työkalu kiinteistöautomaatiojärjestelmissä. Tärkeä työkalu on myös hälytysten hallinta, jolla tarkastellaan järjestelmän toimintaa, ja järjestelmä osaa itse ilmoittaa mahdollisista uhkaavista tilanteista. Järjestelmätapahtumat tarkoittaa toimintoja, jolla voidaan tarkastella yleisellä tasolla, mitä järjestelmässä on tapahtunut ja milloin. Esimerkiksi kuka käyttäjä on vaihtanut pumpun asetusarvoa, mikä oli alkupe-
räinen arvo ja mihin se on muutettu. Turvallisuustoiminnot taas tarkoittavat toimintoja, joilla esimerkiksi jotkin toiminnot on rajattu pois tiettyjen käyttäjien toimialueelta. Käyttöliittymän vaatimukset ovat luonteeltaan ylimalkaisempia kuin aiemmin esitetyt vaatimukset. Pääsyy tähän on se, että esimerkiksi helppokäyttöisyyttä tai selkeyttä on hankala yksilöidä yksittäiseen toimintoon.

Myös tulevaisuuden perusominaisuuksien pohdinta kuuluu tuoterunkoarkkitehtuurin suunnitteluun. Yrityksessä käytyjen keskustelujen ja suoritettujen esitutkimuksen avulla saatiin tulevaisuuden ominaisuuksiksi määriteltyä sääennustepohjainen säätötapa sekä sähkönhintaan perustuva spot-säätö. Kummatkin ominaisuudet liittyvät energianhallintaan, josta yrityksellä on paljon sisäistä osaamista. Myös tulevaisuuden lainsäädännön muutokset tukevat kyseisiä ominaisuuksia. Sääennustepohjaisen säädön ideana on hyödyntää Ilmatieteenlaitoksen tarjoamaa sääennustetta, joka on saatavilla avoimena datana. Sääennusteen avulla voidaan rakennusta lämmittää tai jäähdyttää ennakoivasti. Jos tiedetään, että illalla paistaa aurinko ja on lämmintä, voidaan rakennuksen lämmitystarvetta pienentää jo hieman aiemmin. Sähkön hintaan perustuvalla säädöllä on tarkoitus tuottaa asiakkaalle säästöjä. Säädön toiminta on samankaltainen kuin sääennustepohjaisen säädön toiminta. Yrityksen palvelimelle saadaan joka päivä tieto kuluvan päivän sähkön hinnasta. Hinnaston avulla voidaan kehittää säätöjä, joiden avulla esimerkiksi lämmitetään voimakkaammin, kun sähkö on halpaa.

Ominaisuuksien ongelma on saada eri säätötavat keskustelemaan keskenään toimivasti: niillä on oltava selkeä prioriteettijako ja joustoa säätötavan valinnassa.



Kuva 5.1 Järjestelmän eri säätötapojen on kyettävä keskustelemaan toistensa kanssa.

Kuvassa 5.1 on havainnollistettu suunniteltu järjestelmän säätötapojen hallinta. Tärkeimpänä säätötapana on kalenteriin pohjautuva säätö. Toteutettujen säätötapojen on tarkoitus tehdä kalenteriin pohjautuvasta säädöstä taloudellisesti kannattavampaa ja ympäristöystävällisempää. Kyseisten toteuttaminen on suuri projekti, koska ne joudutaan luomaan tyhjästä. Sen vuoksi ne eivät kuulu tämän työn toteutukseen vaan tulevaisuuden tehtäviin. Näiden ominaisuuksien tuki tulevaisuudessa on kuitenkin otettava huomioon tuoterunkoarkkitehtuurissa.

5.3 Tuoterunkoarkkitehtuurin arkkitehtuurityypin valinta

Kun tiedetään millaisia ominaisuuksia tuoterunkoarkkitehtuuri tarvitsee, on päätettävä miten tuoterunkoarkkitehtuuri toteutetaan. Suurimman ongelman tuoterunkoarkkitehtuurityypin valinnassa tuottaa kiinteistöautomaation perusominaisuuksien ja vaihtelevien ominaisuuksien eli varianssipisteiden suhde. Järjestelmät koostuvat automaatiotasolla samanlaisista osajärjestelmistä. Kenttätasolla nämä automaatiotasoon osajärjestelmät taas koostuvat hyvin erilaisista toimilaitteista ja komponenteista. Asiakkailla voi olla hyvin erilaisia toiveita järjestelmän toteuttamisesta. Alalla on myös muutospaineita kovan kilpailun ja lisääntyvästä lainsäädännön vuoksi. Tulevaisuudessa automaation tarve ja samalla erilaiset tuotteet ovat lisääntymässä. Eri toimijoiden on pakko erottua jollain tavalla massasta. Tuoterunkoarkkitehtuurin toteutustavan valinnassa huomioonotettavat seikat ovat, miten arkkitehtuuri tukee järjestelmän päätoimintaperiaatteita, miten järjestelmän eri osat keskustelevat toistensa kanssa, miten järjestelmä kommunikoi ulospäin ja miten arkkitehtuurin kokoa ja monimutkaisuutta hallitaan.

Tuoterunkoarkkitehtuuri toteutetaan kahdella eri ohjelmistolla. Toteutusohjelmistoilla – TwinCat2:lla ja Indusoftilla – on sisäänrakennettu kommunikointirajapinta keskinäiseen kommunikointiin, mikä helpottaa kahden eri ohjelmiston käyttämistä tuoterunkoarkkitehtuurissa. Toisaalta kaksi erillistä tuoterunkon osaa tuottaa ongelman tuoterunkoarkkitehtuurin ylläpidolle: kahden ohjelman ylläpito on aina hankalampaa kuin yhden.

Yleisellä tasolla kiinteistöautomaattijärjestelmän tuoterunkoarkkitehtuurin ensimmäinen toiminnollinen vaatimus on riittävä muunneltavuuden mahdollistaminen. Arkkitehtuurin avulla on pystyttävä vastaamaan asiakaskohtaisiin vaatimuksiin ja muutuviin markkinoihin. Muunneltavuuden huomioiminen on myös sikäli oleellinen vaati-

mus, että harvoin ensimmäinen versio tuoterunkoarkkitehtuurista on täydellinen – varsinkin, kun kokemus alalta on vähäinen. Tuoterunkoarkkitehtuurin jatkokehittäminen täytyy olla mahdollista.

Muita yleisellä tasolla asettuja vaatimuksia järjestelmälle ovat luotettavuus ja uudelleenkäyttö. Luotettavuus on itsestään selvä vaatimus. Luotettavuuden vaatimukseen vastaaminen tapahtuu laadukkaalla ohjelmoinnilla, suunnittelulla ja ohjelmointisääntöjen noudattamisella. Uudelleenkäyttö taas on koko tuoterunkoarkkitehtuurin peruslähtökohta: tavoitteena on tuottaa ohjelmapohja, jonka päälle voidaan tulevaisuudessa toteuttaa kiinteistöautomaatioprojekteja. Jotta muunneltavuuden vaatimus saadaan toteutettua, arkkitehtuurista ei voida tehdä tiukkaa muuttia. Muunneltavuudessa on pakko ottaa huomioon, miten tuoterunkoarkkitehtuurin sisäisten komponenttien muuttumiseen vastataan. Käytännössä tämä tarkoittaa tuoterunkoarkkitehtuurin eri osioiden välistä kommunikointia ja rajapintoja. Jos riittävää muunneltavuutta ei onnistuta saavuttamaan, ei myöskään arkkitehtuurin uudelleenkäyttö tule olemaan mahdollista..

Tuoterunkoarkkitehtuuri toteutetaan kerrosarkkitehtuurina, minkä tarkoituksena on helpottaa arkkitehtuurin rakenteen jäsentämistä. Kerrosrakenteisuudella vastataan kysymyksiin siitä, miten kiinteistöautomaatiojärjestelmät koostuvat: on pysyviä suuria kokonaisuuksia, joiden sisällä on varianssia. Toisaalta on olemassa kokonaisuuksia, jotka pysyvät oikein suunniteltuna kohtuullisen muuttumattomina kuten PLC:n kommunikointi ulospäin. Kerrosarkkitehtuurissa jokaiselle kerrokselle annetaan oma erillinen tehtävänsä. Kerrokset keskustelevat keskenään rajapintojen avulla. Kuvassa 5.2 on esitetty suunniteltu kerrosrakenne tuoterunkoarkkitehtuurille.

Käyttöliittymä
Osajärjestelmät
Kommunikaatio
Alustus

5.2 Kiinteistöautomaatiojärjestelmän tuoterunkoarkkitehtuuri on kerrosarkkitehtuuri, jolla pyritään helpottamaan kokonaisuuden hallintaa.

Kerrosrakenteen alimmalla tasolla suoritetaan ohjelman alustustoiminnot, eli toiminnot joita tarvitaan ohjelman käynnistymiseen. Esimerkiksi osa TwinCat2:n kirjastoista vaatii alustustoimenpiteitä. Kommunikaatiokerrokselle rakennetaan toiminnot, joiden avulla ohjelma keskustelee ohjattavan järjestelmän kanssa. Käytännössä tämä tarkoittaa anturitietojen lukemista ja ulostulokäskyjä. Osajärjestelmäkerros on suurin kerrosarkkitehtuurin osaa. Se kuvaa kiinteistöautomaatiojärjestelmän eri automaatiotason osioiden toteuttamista. Käytännössä siinä toteutetaan tuoterunkoarkkitehtuurin perusominaisuudet, jotka on listattu taulukossa 5.1. Osajärjestelmäkerroksessa tapahtuu siten myös ohjelman suurin muunneltavuus. Tuoterunkoarkkitehtuurin ylimpänä kerroksena on käyttöliittymä.

Kerrosrakenteen tarkoituksena on sekä selkeyttää arkkitehtuuria että myös tehdä siitä joustavampi: yhden kerroksen muokkaaminen ei suoraan vaikuta muihin kerroksiin. Näin ollen muutos ohjelman yhdessä osiossa ei tarkoita koko ohjelman muokkamista. Kerrosarkkitehtuuri ei kuitenkaan itsessään vielä takaa kovin suurta muokattavuutta: ainoastaan rakenteita, joiden sisäinen muokkaaminen ei vaikuta muihin kerroksiin. Kerrosarkkitehtuurilla muutokset saadaan rajatummiksi. Muunneltavuuden joustavuutta lisätään rakentamalla arkkitehtuurille komponentteja, joiden uudelleenkäyttö ja -määrittely tuoterunkoarkkitehtuurin sisällä on helppoa. Komponentit luodaan modulaarisina black-box-komponentteina, eli ohjelmoijan ei tarvitse tietää miten komponentit on rakennettu – ainoastaan hyödyntää niiden rajapintoja. Käytännössä komponentit vastaavat kiinteistöautomaatiojärjestelmään liitettyjä toimilaitteita.

Kuvassa 5.3 on käyttöliittymän kerrosarkkitehtuuri, joka myös yksi kerros PLC:n arkkitehtuurista. Käyttöliittymän pitää myös pystyä vastaamaan muunneltavuuden ongelmiin. Siksi käyttöliittymän tuoterunkoarkkitehtuurille suunnitellaan samalla tavalla oma kerrosarkkitehtuuri.

Visualisointi
Toiminnollisuudet
Kommunikaatio

5.3 Kiinteistöautomaatiojärjestelmän käyttöliittymän tuoterunkoarkkitehtuuri koostuu kerrosrakenteesta, jolla pyritään helpottamaan kokonaisuuden hallintaa.

Arkkitehtuurin alin kerros on kommunikaatiokerros, joka tarkoittaa erityisesti käyttöliittymäsovelluksen kommunikaatiota PLC-ohjelman kanssa. Toiminnollisuudet kerrokseen kuuluu taulukon 5.3 Yleiset-sarakkeen asiat eli trendikäyrät, hälytystoiminnot ja käyttäjien hallinta. Ylin kerros on käyttöliittymän visualisointi, joka ei liity kooditason toteutukseen vaan ulkonäköön. Ulkonäkö on haastattelujen perusteella tärkeä ominaisuus käyttöliittymälle, joten sen merkitystä on korostettu ottamalla ulkonäkö huomioon myös kerrosarkkitehtuuria suunniteltaessa. Kerrosarkkitehtuurin lisäksi myös käyttöliittymän arkkitehtuuriin kuuluvat modulaariset black-box-komponentit. Näillä komponenteilla on tarkoitus toteuttaa PLC:n tuoterunkoarkkitehtuurin black-box-komponenteille käyttöliittymät.

Valittu strategia tuoterunkoarkkitehtuurin kehittämiseen on luvussa 3.2 esitetyn Kruegerin ohjelmiston sarjatuotannon parannuksen mukainen. Yksittäiset tuotteet eivät ole yhtenäisen muutostenhallinnan kohteena. Pääasiallinen kehitystyö on ohjelmapohjan kehittäminen. Mitä yhtenäisempi ja paremmin suunniteltu tuoterunkoarkkitehtuuri on, sitä paremmin se tukee yksittäisiä tuotteita, jotka elävät omaa elinkaartaan.

Molemmat kehitettävät tuoterunkoarkkitehtuurit ovat siis luonteeltaan kerrosarkkitehtuureja, joiden avulla pyritään kokonaisuuden jäsentämiseen ja parempaan hallintaan. Kerrosrakenteisuus ei pelkästään riitä kokonaisuuden hallintaan: se tarjoaa muunneltavuuteen ja varianssipisteille hyvät olosuhteet. Tärkeä vastaus muunneltavuuteen

teen on toimilaitteille ja eri osajärjestelmille luotavat uudelleenkäytettävät komponentit, joita voidaan monistaa ja hyödyntää järjestelmän sisällä. Komponentit toimivat siten, että ne toteuttavat tietyn rajapinnan. Näin voidaan esimerkiksi korvata yhdenlainen pumppumalli toisenlaisella pumpulla. Komponenteilla voidaan luoda erilaisia suurempia kokonaisuuksia. Komponentit toimivat varianssipisteiden muokkaajina. Varianssipisteet taas sisältävät kerrosarkkitehtuurin kerroksien sisälle. Käytännössä modulaaristen komponenttien hyödyntäminen tarkoittaa komponenttikirjaston luomista molemmalle tuoterunkoarkkitehtuurille.

5.4 Tuoterunkoarkkitehtuurin toteutusvaihe

Tuoterunkoarkkitehtuurin toteutusvaiheen ensimmäinen tehtävä on kiinnittää huomiota ohjelman laatuvaatimuksiin. Ohjelmoinnin laatuun on kiinnitettävä huomiota mahdollisimman aikaisessa vaiheessa. Myöhempi laadun parantaminen on vaikeaa, jopa mahdollonta. Laadun takaamiseksi tuoterunkoarkkitehtuurille suunnitellaan ohjelmointisäännöstö, jossa kerrotaan yksityiskohtaisesti millaisia nimeämiskäytäntöjä tuoterungossa käytetään, miten hoidetaan dokumentointi, mitä mihinkin eri arkkitehtuurin kerrokseen kuuluu ja miten arkkitehtuuriin voidaan lisätä komponentteja tai toimintoja. Ohjelmointisäännöt ottavat kantaa myös rakenteisiin, kommentointiin, koodin ulkonäköön ja yleiseen selkeyteen. Tarkoituksena ei ole ainoastaan siisti ohjelmakoodi vaan ohjelmointiprosessin standardointi. Hyvin todennäköisesti kiinteistöautomaatioprojekteja tullaan tekemään useassa eri yrityksen toimipisteessä ja eri ihmisten toimesta. Standardoidun ohjelman käyttäminen on ohjelmoijalle helpompaa. Tällöin hän tietää miten muuttujat nimitään, mistä eri osioiden toiminnot löytyvät ja niin edelleen. Standardoinnilla on mahdollista nopeuttaa ohjelmointityötä ja parantaa järjestelmän ylläpidettävyyttä. Koska PLC:n ja käyttöliittymän arkkitehtuurit toteutetaan eri ohjelmistoilla, luodaan tätä varten arkkitehtuurin osioille erilliset ohjelmointisäännöt.

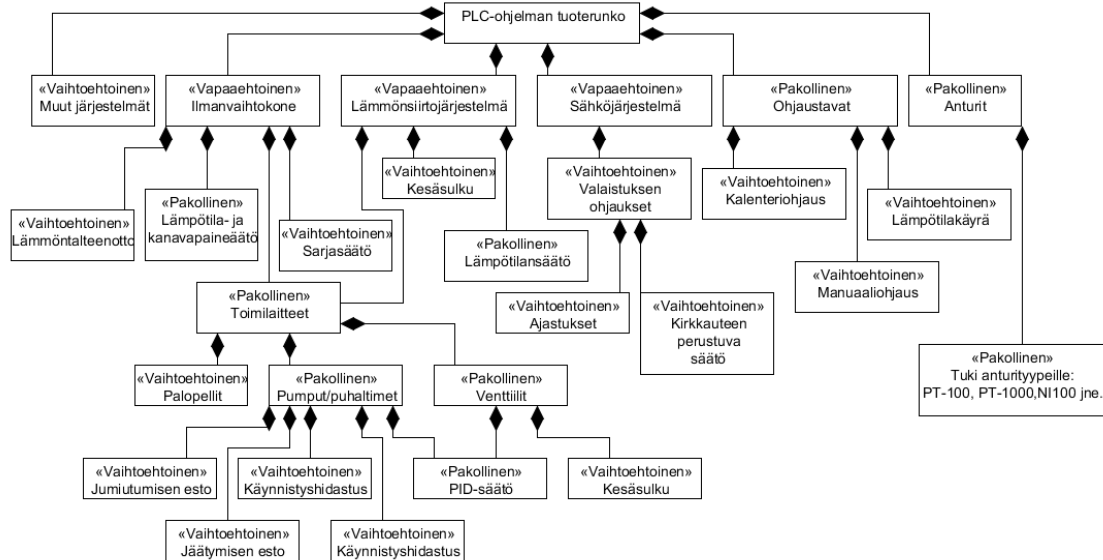
PLC:n tuoterungon ohjelmointisäännöistä otetaan kantaa erityisesti siihen, millainen on ohjelman rakenne. Kerrosarkkitehtuurin ymmärtäminen helpottuu, kun ohjelman eri osiot on selkeästi jaoteltu. Lisäksi säännöissä kerrotaan, miten muuttujat nimitään. Bool-tyyppinen muuttuja nimitään lisäämällä kirjan b muuttujan eteen: bMuuttuja. Jokaiselle muuttujatyypille on oma kirjain tai kirjainyhdistelmä. Nimeämisohjeiden tarkoituksena on helpottaa ohjelman ymmärtämistä – ohjelmoijan ei tarvitse tarkistaa muuttujan alustustaulukosta muuttujan tyyppiä. Säännöissä kerrotaan myös miten ohjelman modulaarisia komponentteja voidaan luoda hyödyntämällä valmiita kirjastoja ja mitä ovat valmiiden komponenttien rajapinnat. Valmiiden komponenttien sisäisiin toimintamekanismeihin ei oteta kantaa, sillä TwinCat2 tarjoaa hyvät kommentointiominaisuudet ohjelman sisällä.

Käyttöliittymän tuoterungon ohjelmointisäännöt ovat luonteeltaan hyvin samanklaiset kuin PLC:n tuoterungon. Niissä otetaan kantaa, miten komponentteja tehdään ja miten niitä käytetään. Indusoft-ohjelmisto ei kuitenkaan tarjoa hyviä ohjelman sisäisiä kommentointimahdollisuuksia. Kommentteja ei voida asettaa suoraan graafisiin kom-

ponentteihin. Ainoastaan komponenttien sisäiset skriptit on mahdollista kommentoida normaalien käytänteiden mukaisesti. Tästä syystä käyttöliittymän ohjelmointisääntöihin on kirjattu jokaisen luodun komponentin toimintaselostus, eli käyttöliittymälle luodut komponentit on siis kommentoitu erilliseen tiedostoon. Verrattuna PLC:n tuoterungon ohjelmointisääntöihin, käyttöliittymän ohjelmointisäännöissä käydään tarkemmin läpi toteutusohjelmiston ominaisuuksia, koska näitä ominaisuuksia hyödynnetään käyttöliittymän arkkitehtuurissa paljon. Uuden ohjelmoijan on nopeampia aloittaa ohjelman käyttäminen, kun toiminnollisuudet ovat tiedossa.

5.4.1 Ohjelmoitavan logiikan tuoterunko

PLC-ohjelman tuoterunko jakautuu edellä tehdyn suunnittelun mukaisesti erilaisiin osiin, joista osa on kiinteitä, eli pakollisia ominaisuuksia tuoterungolle, ja osa vaihtoehtoisia ominaisuuksia. Kuvassa 5.4 on esitetty PLC-ohjelman tuoterungon rakenne piirremallin avulla.



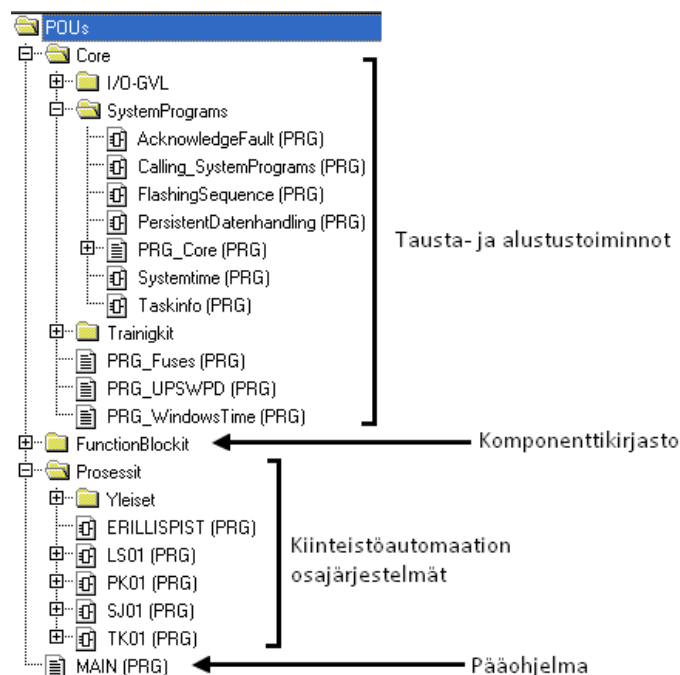
Kuva 5.4 Piirremallin avulla esitetty kuvaus PLC-ohjelman tuoterungon rakenteesta. Pakolliset ominaisuudet ovat tuoterungon kiinteitä ominaisuuksia ja vaihtoehtoiset ominaisuudet tuoterungon varianssipisteitä.

Kuvassa esitetyt järjestelmät ovat karkeasti ottaen sitä suurempia mitä ylempänä ne kuvassa ovat. Rakenne mukailee taulukoiden 5.1 ja 5.2 vaatimuksia. Käytännössä tuoterunkoa jakautuu siis isompiin osajärjestelmiin, jotka jakautuvat yhä pienemmiksi osiksi. Pakolliset ominaisuudet ovat tuoterungon kiinteitä ominaisuuksia ja vaihtoehtoiset varianssipisteitä. Isommat osajärjestelmät ovat kaikki varianssipisteitä, paitsi järjestelmän laitteiden ohjaustavat ja antureiden toimintaan liittyvät asiat. Nämä osiot ovat pysyviä, koska tuoterungon toiminta rakentuu hyvin paljon näiden toimintojen varaan. Säätö tarvitsee aina antureilta saatava informaatiota ja ohjaustavat kertovat, millä perusteella järjestelmää ohjataan. Yleisesti ottaen tuoterungossa on hyvin paljon vaihtoehtoisia ominaisuuksia eli sen avulla voidaan mukautua hyvin asiakaskohtaisiin vaatimuksiin. Erityisesti toimilaitteilla on paljon enemmänkin vaihtoehtoisia ominaisuuksia, kuten

esimerkiksi valinta mikä on pumpun säätimen toimisuunta, mutta ne eivät ole ohjelman rakenteen kannalta niin merkittäviä, että ne kannattaisi piirremallissa esittää.

On kuitenkin selvää, että myös työssä luotujen komponenttien kohdalla varianssinsietokyky tulee väistämättä vastaan. Tämä tarkoittaa sitä, että luotu komponentti ei enää pysty vastaamaan muutokseen. Tällöin vaihtoehtona on, joko uudelleen kehittää komponentti tai luoda kokonaan uusi komponentti tuoterunkoon. Tuoterunko on kuitenkin kehitetty siten, että uusia komponentteja on helppo tuoda tuoterunkoon. Kunhan komponentin tarjoamat liityntärajapinnat ovat tuoterungon vaatimusten mukaiset, ei komponentin sisältämällä koodilla ole merkitystä.

PLC:n tuoterunkoarkkitehtuurissa hyödynnetään standardin IEC-61131 ohjelmointikieliä. Ohjelma jakaantuu pääohjelmaan, siinä kutsuttaviin aliohjelmiin ja aliohjelmissa kutsuttaviin toimilohkoihin. Pääohjelma on rakennettu structured text (ST)-kielellä. Aliohjelmissa on käytetty Function Block diagram (FBD) -kieltä, koska graafisena kielenä se tarjoaa hyvän kokonaiskuvan aliohjelman rakenteesta – suurin ohjelmointityö on tehty aliohjelmien sisällä. Aliohjelmien sisältämät toimilohkot on rakennettu pääasiassa ST-kielellä, koska se tarjoaa monipuolisemmat ja joustavammat ohjelmointimahdollisuudet kuin graafiset ohjelmointikielet. Esimerkiksi silmukkarakenteiden toteuttaminen on helpompaa ST-kielellä. Tuoterunkoarkkitehtuurin kerrosmallin toteutus esitetään kuvassa 5.5.



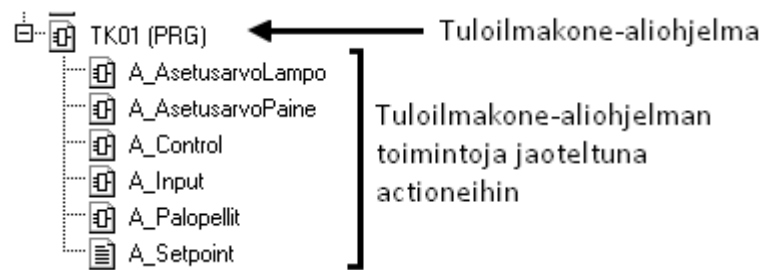
Kuva 5.5 PLC-ohjelman tuoterunkoarkkitehtuurin rakenne. Ohjelman organisoitu kansioihin selventämään rakennetta.

PRG tarkoittaa aliohjelmaa, joka käynnistyy silloin kun niitä kutsutaan pääohjelmassa eli MAIN(PRNG):ssä. Aliohjelmien avulla on rakennettu kiinteistöautomaatiojärjestelmän eri osiot. SystemPrograms kansioon on rakennettu arkkitehtuurin tausta- ja alustustoiminnot, eli kyseessä on PLC:n tuoterunkoarkkitehtuurin alin kerros. Käytännössä

tämä tarkoittaa järjestelmän kellon ajan tasalla pitämistä, erilaisten yleisten toimintojen kuten virheilmoitusten kuittauksia ja niitä toimintoja, joita TwinCat2:n kirjastot vaativat toimiakseen. Edelliseen liittyy esimerkiksi PersistentDatenHandling (PRG). Function-Blockit kansioon on kerätty kaikki arkkitehtuurille luodut komponentit, joita voidaan monistaa ja uudelleen käyttää. Prosessit kansiota löytyy kiinteistöautomaatiojärjestelmän eri osajärjestelmät. LS tarkoittaa lämmönsiirtojärjestelmää, PK poistoilmakonetta, SJ sähköjärjestelmiä ja TK tuloilmakoneita.

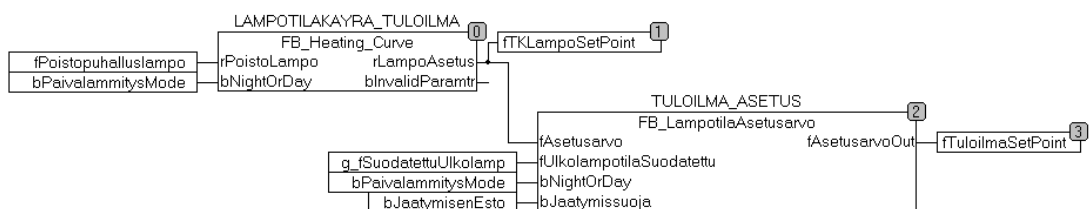
Ohjelmien välisinä rajapintoina toimivat rajapintamuuttujat. Aliohjelmat on jaoteltu ohjelmointisääntöjen mukaisesti kansiorakenteeseen, jolla ei ole ohjelman suorituksen kanssa mitään tekemistä. Tehtävänä on selkeyttää, visualisoida rakenne ja sitä myöten helpottaa ohjelmointia ja ylläpitotoimintoja.

Kuvassa 5.6 on esitetty miten tuoterunkoarkkitehtuurin aliohjelmien eli osajärjestelmien ohjelmointityö on tehty. Esimerkkinä toimii tuloilmakone.



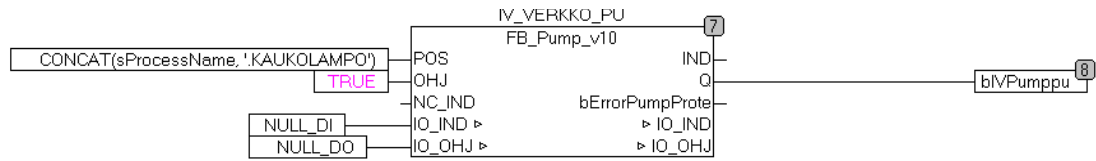
Kuva 5.6 Kiinteistöautomaation tuoterunkoarkkitehtuurin osajärjestelmien ohjelmallinen rakenne. Esimerkkinä tuloilmakoneen ohjelmallinen rakenne.

Kuvassa on siis aliohjelma TK01 (PRG), johon liittyy myös aliohjelman alla listatut toiminnot eli actionit. Aliohjelmaan on tehty toimilaitetason ohjelmointi esimerkiksi pumppujen, puhaltimien ja venttiilien ohjaukset arkkitehtuurisuunnitelmien mukaisesti. Selkeyden vuoksi myös aliohjelma on jaettu osiin käyttäen hyödyksi TwinCat2:n action-toimintoa. Actioneissa määritellään osajärjestelmän – tässä tapauksessa tuloilmakoneen – toiminnan kannalta välttämättä toiminnot: säätimien asetusravot, kalenterin tai muun asetetun toiminnan mukaiset ohjausperusteet, ulkoiset inputit – käytännössä anturitietojen luku – ja palopeltien toiminta. Kuvassa 5.7 on esimerkki tuloilmakoneen tulolämpötilan asetusravon määrittämisessä. Kyseinen ominaisuus on kuvassa 5.6 nimeltään A_asetusarvoLampo.



Kuva 5.7 Kiinteistöautomaation tuoterunkoarkkitehtuurin tuloilmakoneen sisäänpuhalluslämpötilan asetusravon määrittäminen

Actionien sisältämä koodi on toteutettu samalla tavalla kuin muu aliohjelman koodi. Useimmissa tapauksissa on käytetty FBD-kieltä, mutta joskus myös muita IEC-61131 mukaisia kieliä. Itse aliohjelma koostuu actioneista ja black-box-komponenteista. Kuvassa 5.8 esitetään esimerkki toimilaitteen black-box-toteutuksesta.



Kuva 5.8 Kiinteistöautomaation tuoterunkoarkkitehtuurin kaksitoimisen pumpun modulaarinen black-box-komponentti.

Kuvassa on kaksitoimisen pumpun modulaarinen komponentti. Se on rakennettu FBD-kielellä, koska se tarjoaa hyvän ja selkeän rajapinnan inputeille ja outputeille. Komponentin sisäinen koodi taas on toteutettu ST-kielellä. Komponentin vasemmalla puolella ovat sen inputit. CONCAT komennolla toimilaitteelle annetaan ohjelman sisäinen nimi. OHJ tarkoittaa ohjausta. Tässä tapauksessa pumppu on aina päällä. NC_IND tarkoittaa valintaa, mikä toimilaitteen tilatieto on kun tilatietosignaali on 1 tai 0. Jos valinta on 1, niin tilatietosignaalin ollessa 1 myös tilatieto on 1. Jos valinta on 0, on tilatieto 0, tilatietosignaalin ollessa 1. IO_IND inputiin ohjelmoija liittää toimilaitteen tarjoaman tai anturilta saadun digitaalisen tilatiedon ja IO_OHJ inputiin digitaalisen ohjaustiedon. IO on lyhenne sanasta input-output eli tulot ja lähdöt. Lohkon oikealla puolella on sen outputit, joista saadaan pumpun tilatieto (IND), ulostulo (Q) ja virheilmoitukset. Kyseinen komponentti tukee käytännössä kaikkia kaksitoimisia pumppuja, jotka on mahdollista liittää Beckhoffin järjestelmään. Sama asia on voimassa myös muissa luoduissa komponenteissa. Vaatimus laajasta tuesta erilaisille toimilaitteille riippuu tämän lähestymistavan jälkeen siis enemmän järjestelmän rautapuolesta.

Samanlaisella toteutuksella on toteutettu kaikki tuoterunkoarkkitehtuurin black-box-komponentit. Periaatteessa ohjelmoijan tarvitsee tietää vain mitä inputeihin ja outputeihin tarvitsee syöttää. Komponenttien sisältämää koodia on mahdollista muokata ja niistä voidaan luoda uusia komponentteja sovelluksen tarpeisiin. Tämä vastaa arkkitehtuurisuunnitelman modulaarisuustavoitetta.

Kommunikaatio arkkitehtuurin eri osioiden välillä tapahtuu rajapintamuuttujien avulla. Tämä on toteutettu tekemällä sellaisista muuttujista globaaleja muuttujia, joita tarvitaan tuoterungon useassa eri osiossa. Tärkeimmät globaalit muuttujat ovat toimilaitteilta saadut input- ja output-tiedot eli IO_IND ja IO tiedot. Tiedot luetaan PLC:n fyysisellä IO-kortilla. Kortteja on erilaisiin tarkoituksiin, esimerkiksi analogia- ja digitaali-tiedoille omat korttinsa. Toimilaitteiden IO-tiedot on tallennuttu muuttujaan kuvan 5.9 osoittamalla tavalla.

```

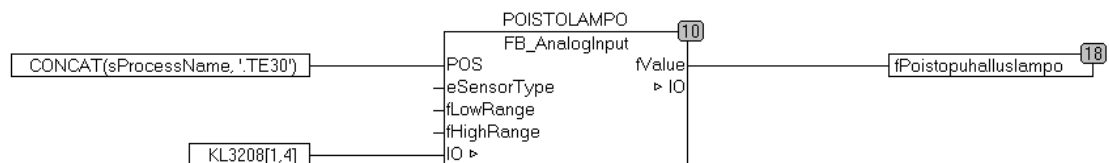
0001 VAR_GLOBAL
0002
0003 (*Määritellään tarvittavat IO-kortit taulukko muodossa. Ensimmäinen taulukko kertoo korttien lukumäärän, seuraava taulukko kertoo IO pisteiden lkm:n*)
0004 KL9210: ARRAY [1..1, 1..1] OF IO_KL92x0; (* 24VDC *)
0005
0006 KL1808: ARRAY [1..1, 1..8] OF IO_DigitalInput;
0007 KL3208: ARRAY [1..1, 1..8] OF IO_AnalogInput;
0008 KL3468: ARRAY [1..1, 1..8] OF IO_AnalogInput;
0009 KL4408: ARRAY [1..1, 1..8] OF IO_AnalogOutput;

```

Kuva 5.9 Kiinteistöautomaation tuoterunkoarkkitehtuurin input- ja output-tietojen alustaminen globaaliin muuttujaan.

Fyysisen IO-kortin signaali luetaan taulukkoon, joka on nimetty kyseisen kortin nimen mukaan. Taulukon koko määräytyy IO-korttien IO-pisteiden mukaisesti. Ensimmäinen taulukkomäärittely kuvaa kyseisten IO-korttien lukumäärä. Seuraava taulukko taas kyseisen kortin IO-pisteiden määrää. Esimerkiksi, jos järjestelmässä halutaan saada digitaalisen input-kortin KL1808:n neljännen inputpisteen tieto, viitataan siihen kirjoittamalla muuttujaksi KL1808[1,4].

Toinen tärkeä arkkitehtuurin kommunikointikerroksen asia ja samalla arkkitehtuurilta vaadittu perustoiminnallisuus, on anturitietojen lukeminen. Kyseinen toiminto tarvitsee hieman erilaisen lähestymistavan, koska erilaisia antureita on olemassa monenlaisia. Toteutus kuitenkin on samanlainen kuin toimilaitteilla eli FBD:llä toteutettu toimilohko ja sisäinen koodi ST-kielellä. Esimerkki anturitiedon lukemisesta on kuvassa 5.10.



Kuva 5.10 Kiinteistöautomaation tuoterunkoarkkitehtuurin anturi-tiedon lukeminen

Toimilohkolle annetaan tarvittavat tiedot: anturin nimi, anturityyppi, ylä- ja alarajat sekä IO-kortin inputmuuttuja. Tämän jälkeen toimilohko skaalaa analogiasignaalin sisäisen koodin mukaisesti oikeanlaiseksi ja tarjoaa ulostulona anturitiedon reaaliaikana. Toimilohkolla voidaan lukea esimerkiksi PT100, PT200, PT1000 tai NI100 antureita. Kyseinen työkalu tuo paljon sujuvuutta kiinteistöautomaatiojärjestelmän toteuttamiseen – jokaiselle anturityypille ei tarvitse erikseen asettaa määrittelyjä, vaan anturityyppejä voidaan vaihtaa ja käyttää joustavasti. Näin saadaan siis täytettyä vaatimus laajasta anturityyppituesta.

Luotu arkkitehtuuripohja tukee myös suunniteltuja erikoisominaisuuksia muunneltavuuden ja modulaarisuuden ansiosta. Erillisiä modulaarisia komponentteja näille ominaisuuksille ei kuitenkaan ole pohjaan vielä luotu. PLC:n tuoterunkoarkkitehtuuri kattaa sille asetetut perusvaatimukset ja on avoin myöhemmälle kehittämiselle. Avoimuus ja myöhemmän kehittämisen mahdollistaminen on varmistettu siten, että tuoterungon eri osiot ovat riittävän eristyksissä toisistaan. Kommunikointi tapahtuu rajapintamuuttujilla. Näin ollen osioiden lisääminen ja muokkaaminen ei sotke tuoterungon toimintaa, kunhan tarvittavat rajapintaehdot toteutuvat.

5.4.2 Käyttöliittymän tuoterunko

Wonderwaren Indusoft -ohjelmistossa on sisäänrakennettu käyttöliittymä graafisten objektien toteuttamiseen. Ohjelmisto tukee myös skriptejä, joita tehdään visual basic-kielellä. Käyttöliittymän tuoterunkoarkkitehtuurin pohjana toimii ja taulukossa 5.3 esitetyt vaatimukset: käyttöliittymän tulee toteuttaa graafinen ulkoasu PLC:lle toteutetulle tuoterungolle.

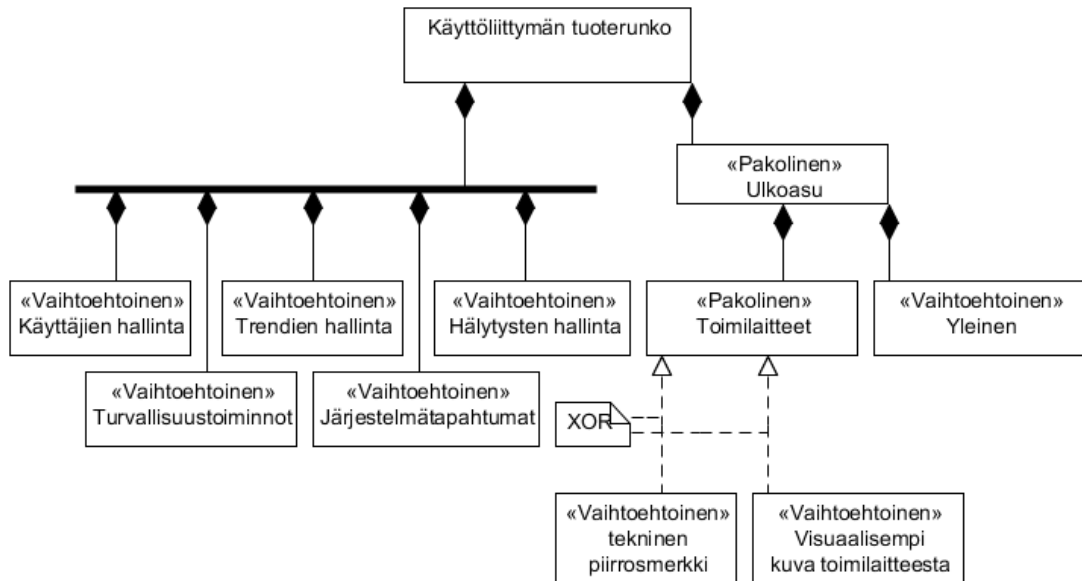
Käyttöliittymä on väylä, jolla ihminen käyttää laitetta tai ohjelmistoa. Suunnittelussa on siis otettava huomioon ihmisen näkökulma. Pelkästään tekniikkaan perustava suunnittelu ei riitä. Suunnittelussa on pyrittävä korkeaan käytettävyyteen, eli siihen, että järjestelmän käyttäjä kykenee käyttämään tuotetta tehokkaasti ja miellyttävästi saavuttaen tavoitteensa. [34]

Käytettävyyden ongelmiin pyritään vastaamaan käyttämällä esitutkimuksessa saatua tietoa alan yleisistä käytännöistä. Suurin haaste on pystyä tarjoamaan kokonaiskuva kiinteistötoiminnasta. Mittauksia, toimilaitteita ja ominaisuuksia on paljon, joten tarjottavaa informaatiota on paljon. Miten tieto tarjotaan kompaktissa ja selkeässä muodossa? Olennainen informaatio pitää olla selkeästi näkyvillä, eikä se saa hukkuu muun informaation alle. Järkevää on hajauttaa tieto segmentteihin: kulutusseuranta, hälytystiedot omiksi kokonaisuuksiksi. [34] Visuaalisuuden tärkeydestä erinomainen osoitus on, että tutkimusten mukaan pelkkä mittarilukeman näyttäminen asukkaalle vähentää kulutusta 10–15 %. [35] Koska suunnitelmana ei ole kuitenkaan tuottaa kotiautomaatiota, ei käyttöliittymästä tarvitse tehdä korostetun selkeää. Järjestelmän käyttäjä tulee todennäköisesti olemaan koulutettu ammattilainen eikä yksityishenkilö. [4]

Kiinteistöautomaation käyttöliittymän tuoterunkoarkkitehtuuri on rakenteeltaan samanlainen kerrosarkkitehtuurin ja modulaaristen black-box-komponenttien kokonaisuus kuin PLC:n tuoterunkoarkkitehtuuri. Kerrosarkkitehtuurimallin toteuttaminen on Indusoftilla helppoa, koska sen sisäiset toiminnot ovat jo valmiiksi jaettu osioihin, jotka voidaan käsittää kerroksina. Esimerkiksi kommunikaatio Beckhoffin laitteiston kanssa on oma sisäinen ominaisuutensa. Samoin trendikäyriille, turvallisuus -ja hälytystoiminnoille on omat sisäiset järjestelmät, jotka ovat suoraan hyödynnettävissä.

Indusoftissa on myös hyödyllinen ominaisuus, jolla voidaan muuttaa luodun käyttöliittymän resoluutiota. Ohjelma skaalaa automaattisesti käytetyt komponentit uuden määrittelyn kokoisiksi. Näin käyttöliittymän tuoterunkoarkkitehtuuria voidaan helposti käyttää eri kokoisten näyttölaitteiden kanssa. Käyttäjien hallintaa varten ei ole suoraan valmista ratkaisua, mutta Indusoft tarjoaa funktiokirjaston, millä käyttäjien hallintatoiminnot ja tapahtumienseuranta voidaan toteuttaa.

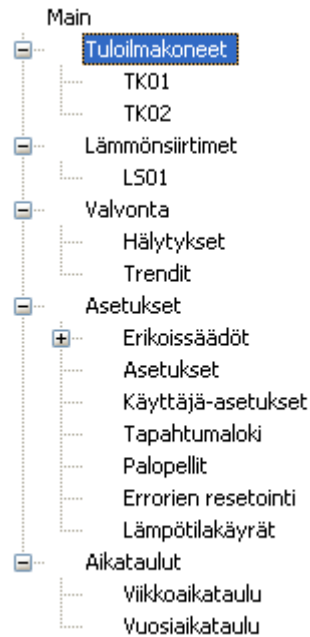
Kuten PLC:n tuoterungossa, myös käyttöliittymän tuoterungossa on omat kiinteät ominaisuutensa sekä varianssipisteensä. Näitä ominaisuuksia on havainnollistettu kuvassa 5.11 piirremallin avulla.



Kuva 5.11 Piirremallin avulla esitetty havainnekuva käyttöliittymän tuoterungon rakenteesta. Pakolliset ominaisuudet ovat tuoterungon kiinteitä ominaisuuksia. Vaihtoehtoiset tuoterungon varianssipisteitä.

Kun käyttöliittymän piirremallia verrataan PLC-ohjelman vastaavaan, havaitaan, että siinä on paljon vähemmän osioita. Yksinkertaisin syy tähän on se, että varsinainen ohjelmointityö tapahtuu PLC:n tuoterungossa. Suurin osa käyttöliittymän tuoterungon modulaarisista komponenteista on niin kiinteästi liittyneenä PLC:n tuoterunkoon, että komponentin varianssin määrittää PLC:n tuoterungon modulaarinen komponentti. Pois lukien komponentin visuaaliset seikat, tärkein varianssipiste onkin päättää, minkälaisia visuaalisia komponentteja käyttöliittymään tahdotaan: teknisiä piirrosmerkkejä vai visuaalisempia kuvia kuten oikeiden pumppujen kuvia. Yleinen ulkoasu viittaa jokaisen käyttöliittymän sivun ulkoasuun. Objektien kokoa, sijaintia ja muita ominaisuuksia on käytetyn ohjelmiston avulla hyvin helppo muuttaa.

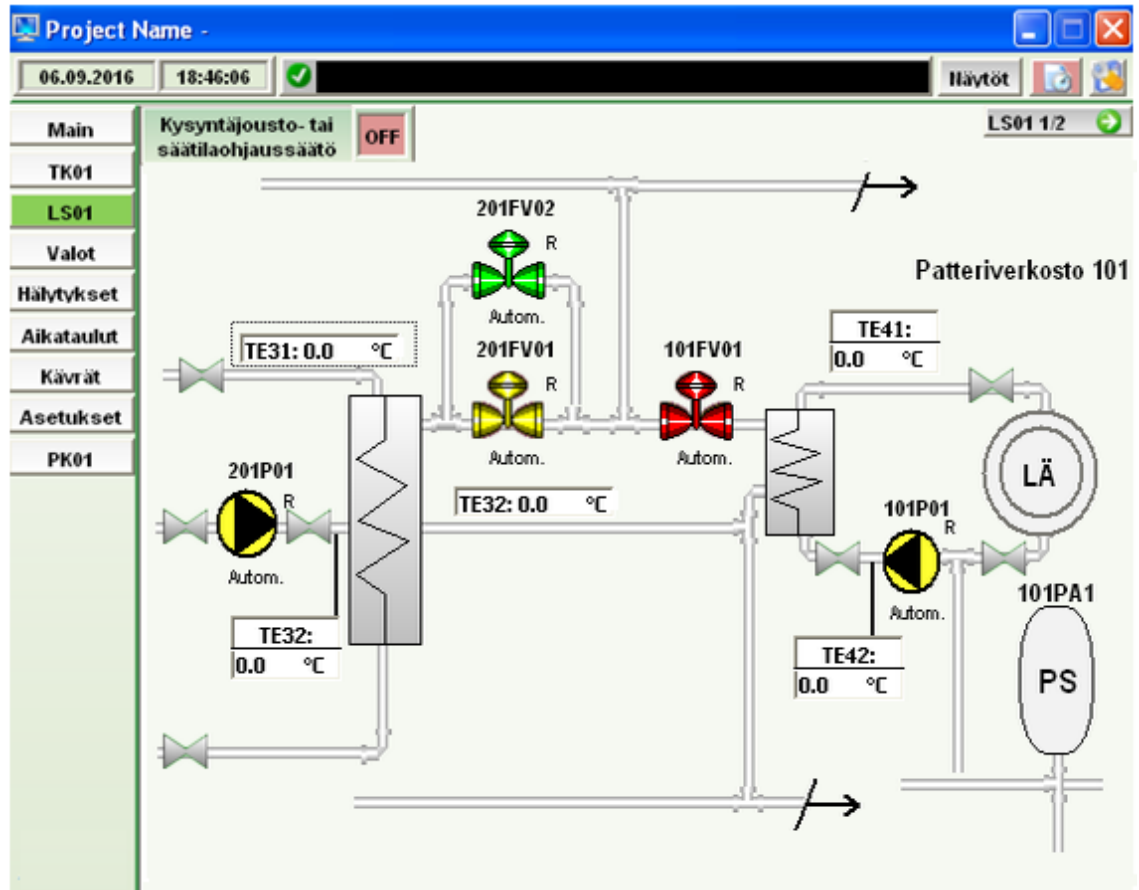
Käyttöliittymän tuoterunkoarkkitehtuurin sovelluksen rakennetta pystyy parhaiten havainnollistamaan tutkimalla miten käyttöliittymä jakaantuu eri sivuihin. Rakenne on esitetty kuvassa 5.12.



Kuva 5.12 Kiinteistöautomaatiojärjestelmän tuoterunkoarkkitehtuurin käyttöliittymän rakenne.

Käyttöliittymän tuoterunkoarkkitehtuuri jakaantuu samanlaisiin osioihin kuin PLC:n tuoterunkoarkkitehtuuri: ilmanvaihtokoneille, lämmönsiirtojärjestelmille ja muille osajärjestelmille on omat osionsa. Lisänä PLC:n tuoterunkoarkkitehtuurin nähdessä on omat osionsa hälytyksille, trendi-käyrille ja muille käyttöliittymässä näytettävillä toiminnoilla.

Kuvassa 5.13 on havainnollistettu käyttöliittymän perusnäkyvä. Havainnollistavana esimerkkinä toimii lämmönsiirtojärjestelmän hallintasivu. Käyttöliittymän perusnäkyvän aina näkyvissä olevat osiot ovat, vasemmalla oleva navigointipalkkialue ja yläreunassa oleva otsikkoalue. Keskelle jäävä suurin alue on alue mihin käyttöliittymän eri sivut tulevat näkyviin. Näin taataan se, että käyttäjällä on jokaiselta sivulta mahdollista päästä helposti muihin käyttöliittymän osioihin.

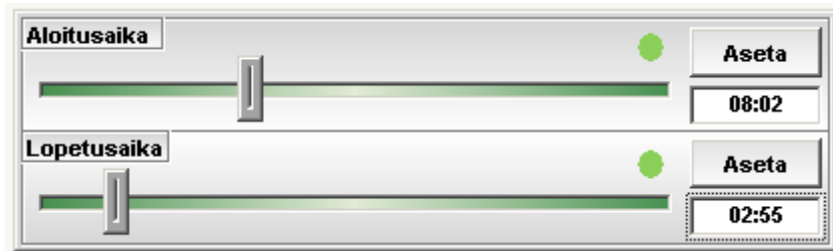


Kuva 5.13 Käyttöliittymän perusnäkökulma. Vasemmalla on pysyvä navigointipalkki ja yläreunassa pysyvä otsakealue. Keskellä on muuttuva näyttöosio, jossa on tässä tapauksessa lämmönsiirtojärjestelmän osajärjestelmän visualisointi.

Haastattelujen ja muiden lähteiden mukaan tärkeimmiksi koetut osiot on sijoitettu käyttöliittymässä aina näkyville. Kaikista tärkeimmät yksittäiset toiminnot on sijoitettu otsakealueelle, jossa näkyy kellonaika ja päivämäärä, järjestelmän viimeisin hälytys sekä pikanäppäimet kalenteri- ja järjestelmäasetuksiin. Toiminnot, jotka eivät ole päivittäisen käytön kannalta oleellisia – kuten käyttäjien hallinta – on sijoitettu "Asetukset"-valikon taakse, jotta etusivu ei tukkeudu liiasta informaatiosta. Tila on rajallisuutensa vuoksi käyttöliittymän tärkein ja vaikeimmin hallittava resurssi. Käyttöliittymän tuoterungon samankaltaisuus PLC:n vastaavaan jatkuu, kun käyttöliittymässä siirrytään syvemmälle.

Kokonaisuus koostuu visuaalisista komponenteista kuten putkistosta ja toimilaitteille toteutetuista käyttöliittymistä. Toimilaitteiden käyttöliittymät ovat arkkitehtuurin modulaarisia komponentteja. Niiden toteuttamisessa on hyödynnetty Indusoftin linked symbol -ominaisuutta. Kyseisellä ominaisuudella graafisia ominaisuuksia voidaan ryhmittää ja tehdä niistä yksittäisiä, monimutkaisempia komponentteja. Linked symbol -komponentti voidaan käsittää samantapaisena oliona kuin TwinCat2:n toimilohko. Symbolille syötetään sen tarvitsemat muuttujat, eikä ohjelmoijan tarvitse tietää miten kyseinen objekti on luotu tai miten se toimii. Eli myös käyttöliittymän tuoterungossa komponentit näyttäytyvät ohjelmoijalle black-box-komponentteina. Näin ollen komponentteja on helppo muuttaa tai tuottaa lisää – ne toimivat tuoterungossa, kunhan komponentin tarjoamat rajapinnat ovat tuoterungon vaatimalla tasolla. Kuvassa 5.14 on

esimerkki yhdestä luodusta linked symbol -komponentista. Käymällä läpi miten komponentti on rakennettu, saadaan kuva siitä, millaisia ongelmia komponenttien rakentamiseen liittyy.



Kuva 5.14 Erilaisten ajatusten aloitus- ja lopetusajan määrittämiseen käytettävä komponentti.

Kyseisellä komponentilla voidaan määrittää esimerkiksi kalenteriajastuksen aloitus- ja lopetusaika. Aika voidaan määrittää joko liukuvalla palkilla, jolloin oikealla näkyvä kellonaika päivittyy reaaliaikaisesti, tai painamalla kelloaikaa, jolloin näytölle aukeaa näppäimistö, jolla kellonaika voidaan syöttää. Jos aikaa on muutettu, mutta ei asetettu painamalla aseta-nappia, napin vieressä oleva vihreä merkkivalo muuttuu keltaiseksi. Kun aika asetetaan, muuttuu merkkivalo vihreäksi. Merkkivalon tarkoitus on ehkäistä kellonajan asettamisen unohtamista muutoksen jälkeen. Liukukyttimeen arvorajat ovat 0–1439, koska vuorokaudessa on 1440 minuuttia. Kyttimeen arvosta lasketaan valittu tunti yhtälöllä

$$h = \frac{x}{60}, \quad (1)$$

jossa h on tuntien lukumäärä ja x on liukukyttimeen arvo. Minuuttien lukumäärä on yhtälön (1) laskusta saatu jakojäännös

$$m = \text{mod}(h). \quad (2)$$

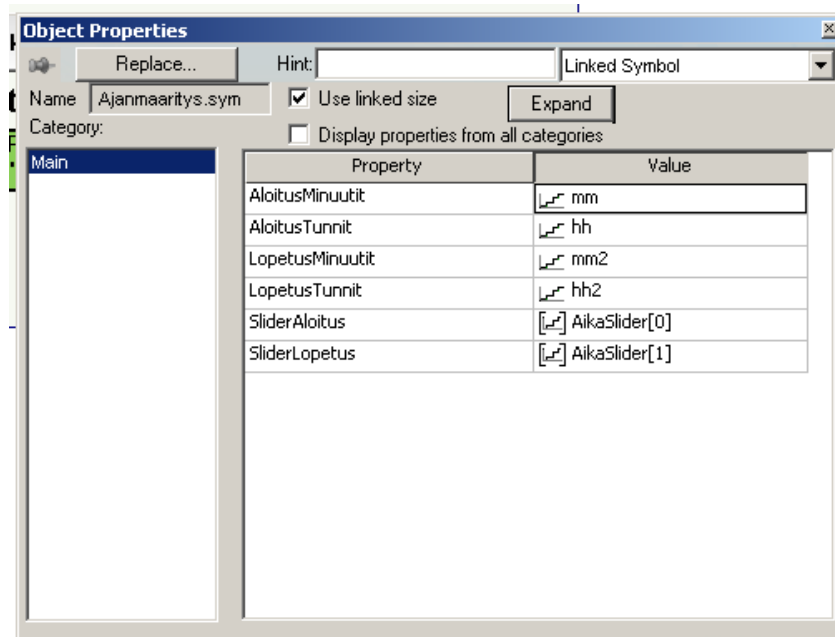
Objektin skripteissä, lasketut tunnit ja minuutit muutetaan näkymään muodossa tunti:minuutit. Kuvassa 5.15 esitetään objektin kellonaikaa painamalla aukeava näppäimistö.



Kuva 5.15 Indusoftin tarjoama kosketusnäyttönäppäimistö.

Kuvasta huomataan, että näppäimistössä ei ole kaksoispistettä. Tämän vuoksi kellonaika syötetäänkin komponenttiin muodossa tunnit.minuutit. Objektin taustalla on skripti, joka toteuttaa virheen tarkistuksen syötetylle kellonajalle. Kellonajan tulee olla oikean muotoinen ja asettua kellonajoille luonnollisiin raameihin. Jos syöte on virheellinen, sitä ei hyväksytä.

Modulaarisissa komponenteissa on pyritty siihen, että iso osa virheentarkistuksesta suoritetaan käyttöliittymän ohjelmassa. Tarkoituksena on estää väärämuotoisten syötteiden joutuminen PLC-ohjelman komponentteihin. Virheellisesti muotoiltu syöte ajaa PLC-ohjelman virhetilaan, josta se täytyy manuaalisesti nollata. On parempi estää virheelliset syötteet jo käyttöliittymän puolella. Kuvassa 5.16 esitetään millainen käyttöliittymä Indusoftin linked symbol -komponenteilla on ohjelmoijalle, kun muuttujia linkitetään komponenttiin.



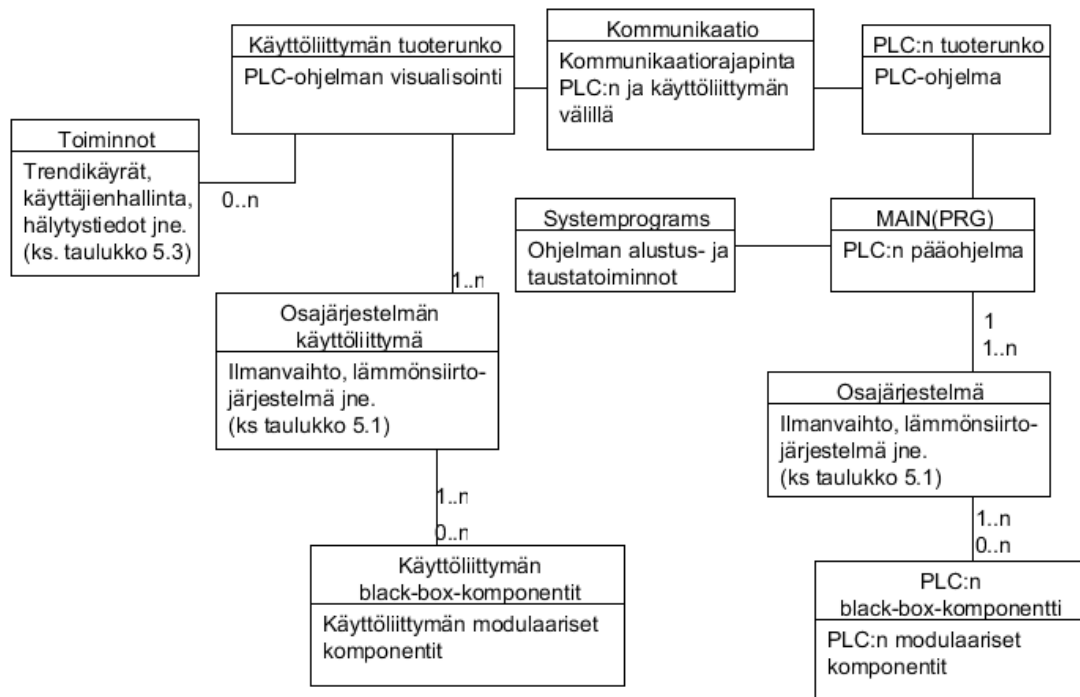
Kuva 5.16 Linked symbol -komponentin käyttöliittymä muuttujien linkitykselle.

Vasemman puoleisessa sarakkeessa on Indusoft-sijaismuuttuja, johon linkitetään oikean sarakkeen PLC-muuttuja. Linked symbol -komponenttia voidaan uudelleenkäyttää kopioimalla se ja linkittämällä siihen uudet muuttujat. Käyttöliittymään on luotu modulaarinen komponentti kaikille PLC-ohjelman toimilaitteille.

Koko käyttöliittymän luomisen ajan on kiinnitetty huomiota myös visuaaliseen ilmeeseen. Ulkonäköseikoille ei kuitenkaan anneta liikaa painoarvoa, koska projektin päätaavoite on suunnitella teknisesti toimiva tuoterunkoarkkitehtuuri. Visuaalisten ominaisuuksien muokkaaminen on mahdollista myös jälkeinpäin. Kun tuotteen kokonaisuus on selvillä, on helpompi tehdä yhtenäinen visuaalinen ilme. Linked symbol -komponentit vastaavat osaltaan myös visuaalisen ilmeen muokkaamiseen: jokaiseen samaan komponenttiin pohjautuvan objektin ulkonäkö on mahdollista määrittää yhteisesti.

5.5 Tuoterunkoarkkitehtuurin luokkakaavio

Tässä luvussa käydään läpi, miten kahdessa edellisessä luvussa muodostetut tuoterungot jakaantuvat eri osiin. Tarkoituksena on havainnollistaa suunnitteluvaiheessa esitettyjä asioita. Tätä on havainnollistettu kuvassa 5.17.



Kuva 5.17 Luokkakaavion avulla havainnollistettu tuoterunkoarkkitehtuurin rakenne

PLC:n ja käyttöliittymän tuoterungot kommunikoivat sisäänrakennetun kommunikointirajapinnan avulla. PLC:n tuoterunko on käytännössä MAIN(PRG), koska siihen on rakennettu kaikki PLC:n kiinteistöautomaation osajärjestelmät ja ohjelman vaatimat taustatoiminnot. Osajärjestelmiä voi olla niin paljon kuin asiakas tahtoo. Nämä toiminnot on lueteltuna taulukossa 5.1. Osajärjestelmät taas voivat koostua modulaarisista black-box-komponenteista.

Käyttöliittymän tarkoituksena on luoda ihmisen ja koneen välinen rajapinta PLC-ohjelmalle. Siihen kuitenkin liittyy myös sisäisiä toimintoja, kuten käyttäjienhallintaa ja hälytystoimintoja. Nämä toiminnot on lueteltuna taulukossa 5.3. Myös käyttöliittymä voi sisältää modulaarisia black-box-komponentteja, joiden avulla tuoterungon osajärjestelmät voidaan luoda.

5.6 Tuoterunkoarkkitehtuurien testaaminen

Ohjelmoidun tuoterunkoarkkitehtuurin testaamisessa nousee esiin esitutkimuksesta tuttu ongelma: kuinka testata järjestelmää, joka ei ole täysin kokonainen vaan tuoterunko. Ongelma korostuu PLC:n tuoterunkoarkkitehtuurin tapauksessa, johtuen TwinCat2:n simulointiympäristöstä. Säädön toimivuus voidaan testata siten, että anturitiedon tilalle syötetään manuaalinen arvo. Sääto alkaa toimia, esimerkiksi pumpun ulostulo kasvaa.

Tällä ulostulolla ei ole kuitenkaan mitään vaikutusta manuaalisesti syötettyyn muuttu-jaan. Toisin sanoen säädön toimivuus voidaan testata sillä tasolla, että toimilatteiden toimitusunnat ovat oikeat. Toisaalta säädön tarkka testaaminen ei ole tässä vaiheessa tarpeellista. Jokainen järjestelmä on joka tapauksessa erilainen ja säätöparametrit täytyy määrittää kiinteistöautomaatiojärjestelmille yksilöllisesti.

Tuoterunkoarkkitehtuurin mahdollisimman kattavaan testaukseen pyritään kuitenkin läpi koko projektin, jolloin virhetilanteisiin on mahdollista reagoida heti. Tällöin saadaan minimoitua virheiden kertaautuminen. Tärkeintä tuoterunkoarkkitehtuurissa on, että sen perustoiminnollisuudet ovat kunnossa: esimerkiksi lukitukset toimivat kuten pitää ja säätösilmukat toimivat oikeaan suuntaan.

Testaamisen puutteellisuuteen olisi voitu puuttua, jos projektin aikana olisi saatu hankittua demoprojekti. Näin tuoterunkoarkkitehtuurin avulla olisi voitu toteuttaa kokonainen järjestelmä, jolla mahdolliset ongelmakohdat olisi saatu esille. Koska demoprojektia ei onnistuttu hankkimaan, jäi tuoterunkoarkkitehtuurin testaaminen pintapuoliseksi. Simuloinnilla, jonka TwinCat2 tarjoaa, ei voida saada selville kaikkia reaali- maailman ongelmia, kuten viiveitä ja ongelmia aiheuttavia toimilaitteita.

Edelliset ongelmat liittyvät erityisesti PLC:lle rakennettuun tuoterunkoarkkitehtuuriin. Myös käyttöliittymän ohjelman testaaminen on kokonaisuuden kannalta yhtä tärkeässä osassa. Sen testaamisessa ei ole kuitenkaan samanlaista ongelmaa kuin PLC:n tuoterunkoarkkitehtuurin testaamisessa. Käyttöliittymän tuoterungossa ei tarvitse testata monivaikutteisia säätörakenteita, vaan toiminnollisuudet ovat rajatumpia. Esimerkiksi sivujen täytyy toimia tietyllä tavalla: tietty informaatio on esillä, kun tietyt ehdot toteutuvat. Samoin linked symbol -komponentit toimivat ikään kuin järjestelmästä erillään, sisäisten ehtojensa mukaisesti, joten niiden toiminta voidaan testata komponenttikohtaisesti. Testaamisessa voidaan käyttää PLC:n arkkkitehtuurin muuttujien tilalla käyttöliittymän sisäisiä muuttujia, jolloin toiminnollisuuksien testaaminen helpottuu entisistään. PLC-ohjelman ei tarvitse olla päällä eikä yhdistettynä käyttöliittymään, kun toimintoja testataan. Indusoftin simulointiympäristö tarjoaa siis hyvät työkalut testaamisen toteuttamiseen.

Vaikka varsinaista demo-projektia ei saatu hankittua, saatiin testaamista varten hankittua PLC ja näyttö. Testilaitteena toimii Beckhoffin CP6207-0001-0020. Laite on PLC, johon on integroitu näyttö. Tuote auttoi erityisesti käyttöliittymän testaamisessa, koska sen avulla saa testattua miltä käyttöliittymä näyttää fyysisellä näytöllä. On esimerkiksi mahdollista toteuttaa käyttäjätestausta: ovatko napit riittävän suuria kosketusnäytön käyttämiseen ja muita vastaavia testejä.

Kuitenkin myös testilaite tuottaa itsessään puutteita testaamiseen. Testilaite ei vastaa niitä komponentteja, joilla kiinteistöautomaatioprojekteja tullaan tulevaisuudessa tekemään. Testilaitteen prosessoriteho ja muisti on huomattavasti alempi kuin projekteihin suunnitellun PLC:n. Näin ollen on vaikea määrittää, kuinka raskaita visuaalisia elementtejä käyttöliittymässä voidaan käyttää, että käyttö pysyy sujuvana.

5.7 Dokumentointi

Kuten testaaminen myös dokumentointi on oleellinen osa projektin toteuttamista sen koko elinkaaren ajan. Pää tarkoituksena on kertoa miten sovellus kokonaisuudessaan toimii ja miten sen eri osiot ja komponentit toimivat. PLC:lle luotu tuoterunkoarkkitehtuuri on helpompi dokumentoida kuin käyttöliittymän tuoterunkoarkkitehtuuri, koska TwinCat2-ohjelmisto tarjoaa kommentoinnille hyvät työkalut. Kommentteja voidaan ohjelmassa laittaa käytännössä mihin tahansa. Ne on helppo upottaa koodin sekaan. Kommenttien lisäksi tästä tuoterunkoarkkitehtuurin rakenteeseen, eri osioiden toimintaan ja arkkitehtuurin tarkoitukseen on otettu kantaa ohjelmointisäännöissä. Ohjelmointisääntöihin kirjataan myös tehdyt rakenteelliset muutokset. Pääasiallinen komponenttien ja toiminnollisuuksien dokumentointi tapahtuu kuitenkin ohjelman sisäisillä kommentteilla.

Käyttöliittymän tuoterunkoarkkitehtuurin dokumentointi on ongelmallisempaa. Indusoft -ohjelmisto ei tarjoa esimerkiksi graafisten objektien toiminnollisuuksien kommentointiin käytännöllistä tapaa. Ohjelmistossa kommentteja voidaan syöttää ainoastaan näyttöjen tai komponenttien skripteihin. Näin ollen kommentit ovat piilossa eikä niitä ole helppo yhdistää komponentin eri ominaisuuksiin. Objekteihin kirjoitetut skriptit kuitenkin kommentoidaan mahdollisimman huolellisesti. Lisäksi jokaiseen käyttöliittymään kuuluvan näytön toiminnollisuus on dokumentoitu kyseisen näytön skriptiosioon. Kuten todettua, tällaista mahdollisuutta ei ole modulaarisilla komponenteilla. Näin ollen modulaaristen komponenttien ja käyttöliittymän tuoterunгон yleistason kommentointi tapahtuu ohjelmointisäännöissä – ohjelmasta erillisessä suunnitteludokumentaatioissa.

6 TULOSTEN ARVIOINTI

Työn tuloksena saatiin luotua yritykselle tuoterunko kiinteistöautomaatiojärjestelmälle ja siten liitettyä kiinteistöautomaatio osaksi yrityksen tuoteportfoliota. Työn onnistumista kokonaisuutena voidaan arvioida käytetyn innovointiprosessimallin avulla. Kokonaisuutena innovointiprosessi oli toimiva. Se jaksotti tuotteen kehittämistä selkeisiin vaiheisiin ja arviointiporttien avulla yrityksen johto pysyi myös mukana tuotteen kehittämisessä. Suurin puute innovointiprosessissa oli kuitenkin asiakaskontaktin puuttuminen, koska tuotetta kehitettiin yrityksen omien valmiuksien parantamiseksi. Tätä puutetta onnistuttiin kuitenkin korjaamaan esi- ja markkinatutkimuksen avulla. Täysin ne eivät asiakaskontaktia korvaa, mutta antavat hyvää yleistietoa markkinoista. Ylivertaisesti paras arvio sekä prosessin että kehitetyn tuoterunkoarkkitehtuurin onnistumisesta saataisiin, jos tuoterunkoa päästäisiin käyttämään asiakasprojektissa.

Prosessissa suoritettiin esi- ja markkinatutkimus, joiden tarkoituksena oli vastata kysymyksiin: millainen on hyvä kiinteistöautomaatiojärjestelmä ja miten nykyisiä kiinteistöautomaatiojärjestelmiä voitaisiin parantaa? Esi- ja markkinatutkimusten tuloksena saatiin tärkeää tietoa siitä, mitä ovat markkinoiden yleiset käytännöt ja kehityskohteet. Tutkimuksista saatujen tietojen pohjalta oman tuotteen toteuttaminen helpottui. Esitutkimuksen avulla saatiin myös varmistettua, että yrityksellä on olemassa valmiuksia suorittaa innovointiprosessi loppuun ja että suunniteltu tuote on sopiva yrityksen omaan strategiaan. Koen, että teemahaastattelut olivat oikea tapa tuoda tärkeää lisäinformaatiota esi- ja markkinatutkimuksiin, koska niiden avulla saatiin tarkempaa ja syvällisempää informaatiota kuin perinteisemmällä lomakehaastattelulla. Jokainen järjestelmä on erilainen ja eri henkilöillä on erilainen asiantuntijapohja. Teemahaastatteluiden avulla oli mahdollista syventyä juuri kyseisen haastateltavan asiantuntijuuteen. Innovointiprosessissa päästiin tuotekehitysvaiheeseen ja prototyypin testaamiseen. Jatkotutkimuksessa olisi tärkeää suorittaa innovointiprosessi loppuun, eli hoitaa loppuun tuotteistaminen, markkinointi, myynti ja lopulta ylläpito. On suunniteltava loppuun tuotepaketit, tehtävä markkinointi materiaalit ja pohdittava millaisena tuotteena kiinteistöautomaatiojärjestelmiä tahdotaan myydä.

Tulevaisuutta kannattaa pohtia myös hallinnolliselta näkökannalta: kuinka paljon henkilöstöresursseja varataan järjestelmää varten, millaisia työvälineitä ja materiaaleja on hankittava. Resurssien tarve riippuu siitä, miten paljon projekteja tavoitellaan. Mitä enemmän projekteja on sitä enemmän on myös ylläpidettävää. Tulevaisuuden työntekijät eivät automaattisesti osaa käyttää järjestelmää, joten myös koulutusmateriaali on tuotettava. Kuten ohjelmistotuotannossa myös hallinnollisella puolella on ajansäästämiseksi järkevää vakioda usein toistuvia toimenpiteitä.

Kuten oletettua, innovointiprosessin aikaa vievin ja hankalin vaihe oli tuotekehitysvaihe, jossa toteutettiin tuoterunkoarkkitehtuuri kiinteistöautomaatiojärjestelmälle. Tuoterunkoarkkitehtuuri jakaantui kahteen erilliseen, mutta yhdessä toimivaan tuoterunkoon: PLC:n- ja käyttöliittymän tuoterunkoihin. Tärkein syy oli, että käyttöliittymän tekemisessä pystyttiin näin hyödyntämään erillistä ohjelmistoa. Koska erillisellä ohjelmistolla oli mahdollista tehdä käyttöliittymän arkkitehtuuri helpommin ja myös näyttävimmän kuin PLC:n ohjelmalla ja silti säilyttää tuoterunkojen yhteistoimivuus, oli kahden erillisen tuoterunkon tekeminen kannattavaa. Tästä kuitenkin aiheutuu ongelmia ylläpidettävyydelle: kahta ohjelmistoa on aina hankalampi ylläpitää kuin yhtä. Toisaalta, vaikka olisi käytetty vain yhä ohjelmistoa, olisi käyttöliittymän ohjelma silti ollut PLC:n ohjelmasta erillinen ja vaatinut erillisiä ylläpitotoimintoja.

Ehkä tärkein kysymys tulosten arvioinnissa on, oliko tuoterunkoarkkitehtuurin rakentaminen kannattavaa verrattuna esimerkiksi perinteisempään tuotekehitykseen. Eli onnistuuko luotu tuoterunkoarkkitehtuuri tuomaan esiin luvussa 3 esitellyt tuoterunkoarkkitehtuurin edut? Koko kysymyksen vastaus kulminoituu tulevaisuuteen. Jos kiinteistöautomaatioalan markkinoille pääsy ei onnistu ja siten kiinteistöautomaatioon liittyviä projekteja ei ole saatavilla ei tuoterunkon kehittäminen ollut kannattavaa. Tuoterunkoarkkitehtuurin kehittäminen vaatii paljon resursseja ja alkukustannuksia – enemmän kuin perinteinen, ainoastaan nykyhetken tarpeeseen vastaava, ohjelmistokehitys. Kuitenkin, jos projekteja on saatavilla, kehitetyn tuoterunkon avulla tuote on nopeammin ohjelmoitavissa ja siten nopeammin saatavissa asiakkaalle. Tuoterunkoarkkitehtuuri standardisoi tuotteen kehitystä ja antaa useita työkaluja ohjelmoijalle. Tässä työssä nämä asiat näkyvät esimerkiksi kehitettyinä modulaarisina komponentteina ja luotuina ohjelmointisääntöinä. Kun projektit luodaan saman tuoterunkon pohjalta samoja ohjelmointisääntöjä noudattaen, on myös projektien ylläpito yksinkertaisempaa. Kunhan tuoterunkoa hyödynnetään oikein ja ohjelmointisääntöjä noudatetaan, tämä tulos on mahdollista saavuttaa kehitetyllä tuoterungolla.

Ohjelmistotasolla tuoterunkoarkkitehtuurin kehittämisen tärkein kysymys on, ovatko tuoteperheen tuotteet riittävän samankaltaisia tuoterunkoarkkitehtuurin kehittämiseksi. Tähän kysymykseen saatiin vastauksia esi- ja markkinatutkimuksen avulla. Kiinteistöautomaatiojärjestelmät ovat ominaisuuksiltaan hyvin samankaltaisia: jokaisessa toistuu ilmanvaihtokone, lämmönsiirtojärjestelmät ja muut suuremmat osakokonaisuudet. Toistuvuudet tukevat tuoterunkoarkkitehtuurin suunnittelemista. Ongelmia syntyy, kun suurempiin kokonaisuuksiin liitetään kokonaisuuDET toteuttavat toimilaitteet. Toimilaitteita on lukematon määrä erilaisia ja jokaisen eri toimilaitteen vaatimuksiin on mahdollon varautua tuoterunkon avulla. Koska jokaisella asiakkaalla ei varmasti ole tiukkoja vaatimuksia käytettävistä toimilaitteista, kehitetyllä tuoterungolla pystytään varmasti vastaamaan suurelta osin eri projektien väliseen varianssiin.

Asiakaskohtaiset muutostarpeet ovat suurin syy, miksi tuoterungossa suurien linjojen standardoinnin sijaan toimilaitteista ja toimilaitteiden tarvitsemista toiminnoista on tehty modulaarisia komponentteja. Näitä komponentteja on mahdollista siirrellä ja käyttää tuoterunkon sisällä. Suuret linjat on tuoterunkoarkkitehtuurissa toteutettu käyt-

tämällä kerrosmallista arkkitehtuuria. Jokaisella kerroksella on oma tehtävänsä ja on erillinen muista kerroksista: keskustelu kerrosten välillä tapahtuu rajapintamuuttujien avulla. Tämä ratkaisu takaa sen, että esimerkiksi ilmanvaihtokoneen toiminta voidaan ohjelmoida asiakkaan tarpeen mukaan, eikä se vaikuta muiden järjestelmän osioiden toimintaan. Tiedot mitä ilmanvaihtokoneesta tarvitaan muihin osioihin, saadaan käyttämällä mainittuja rajapintamuuttujia. Vapaus ja muunneltavuus ovat hyvin tärkeitä asioita tuoterunkoarkkitehtuuria ja suunnitellulla tuoterungolla tämä saavutetaan.

On syytä kuitenkin pohtia, onko tuoterungossa annettu vapaus liian suuri. Jokaiselle suuremmalle osajärjestelmälle on tuoterungossa annettu toteutusmalli, mutta tietenkään sitä ei halutessaan ole pakko noudattaa. Totuttujen rakenteiden muuttuminen voi aiheuttaa ylläpito-ongelmia. Toisaalta tässä vaiheessa tuotekehitystä ei ole tietoa, ovatko nyt suunnitellut suuremmat rakenteet toimivia. Asia selviää vasta, kun tuoterungolla on toteutettu asiakasprojekteja. Muutoksia ja kehitystä tarvitaan varmasti. Tämän vuoksi tuoterungon sisällä annettu vapaus on kuitenkin eduksi. Riittävän vapauden vuoksi esimerkiksi toteutettavan järjestelmän koolla ei ole merkitystä, mikä oli yksi suunnittelun päätavoitteista.

Asiakasprojektit paljastavat lopulta myös sen, kuinka paljon kehitetty tuoterunko nopeuttaa asiakasprojektien tekemistä. Tällä hetkellä nopean projektin tuottamiseen on hyvät mahdollisuudet: toimilaitteille ja toiminnoille on olemassa valmiita, heti käytettäviä komponentteja, tuoterunko antaa pohjan osajärjestelmien suunnittelulle ja ohjelmointisäännöt kertovat miten eri komponentteja ja tuoterungon osioita tulee käyttää.

Itse tuoterungossa tärkeintä on variointipisteiden toteutus, eli miten tuoterunko sietää muutospaineita. Muutospaineet voivat johtua esimerkiksi asiakkaan vaatimuksista tai standardien ja lainsäädännön kehittymisestä. Muutospaineisiin on vastattu modulaarisilla komponenteilla. Tärkeää komponenteissa on, että sekä PLC:n- että käyttöliittymän tuoterunkoarkkitehtuureissa, modulaariset komponentit on toteutettu black-box-komponentteina. Komponenteilla on rajapinta, jota ohjelmoija käyttää. Modulaaristen komponenttien etu on niiden siirreltävyys ja se, että niitä on helppo luoda lisäksi erilaisiin tarpeisiin. Haittapuolena on, komponenttien määrän lisääntyessä, myös niiden ylläpito hankaloituu. Paras keino vastata tähän on komponenttien tarkan dokumentoinnin jatkaminen. Kehitetyt komponentit ovat melko pieniä, koska pienempien komponenttien uudelleenkäyttäminen on helpompaa ja joustavampaa kuin suurien. Suuret komponentit ovat hankalammin ylläpidettäviä ja vaikeammin uudelleenkäytettäviä. Koska komponentit ovat modulaarisia, on erilaisten komponenttien käyttäminen tuoterungossa ja siten projekteissa helppoa ja näin ollen tuoteperheen tuotteiden välisiin muutoksiin ja tulevaisuudessa tapahtuvaan tuotteen kehittymiseen on pystytty varautumaan.

Käyttöliittymän suurimpana kehityskohteena on ulkonäön parantaminen eli yhtenäisen visuaalisen ilmeen suunnittelu. Tällä hetkellä käyttöliittymän tuoterungossa on pyritty yhtenäiseen ja siistiin suunnitteluun, mutta se ei kuitenkaan ollut kehityksen painopiste. Näin olleen ulkonäössä on paljon parannettavaa. Eräs keino parantaa käyttöliittymään ulkonäköä voi olla valmiiden activex-komponenttien hyödyntäminen. Ne ovat yleensä ulkonäöltään moderneja ja sisältävät käytännöllisiä toiminnallisuuksia. Activex-

komponentti on Microsoftin kehittämä konsepti. Käytännössä se on pienoisohjelma, joka mahdollistaa esimerkiksi animaatioiden käyttämisen näytössä. Valmiita Indusoftin tukemia activex-komponentteja löytyy useaan eri tarkoitukseen. [36]

Kuten edellisessä luvussa mainittiin, tuoterungon testaamisen vajavaisuus on tämän työn suurin puute. Testaamisen ongelmat liittyvät tuoterunkopohjaiseen rakentamiseen ja erityisesti siihen, ettei tuoterungolle saatu hankittua demoprojektia. Testaamisen suorittaminen loppuun ja demoprojektin toteuttaminen olisi tärkeää suorittaa jatkotutkimuksena.

Kehitetyn tuotteen onnistumista ei voi pohtia pelkästään alkuperäisten tutkimuskysymysten avulla. Suuri merkitys on sillä, miten tuote vertautuu kilpailijoiden vastaaviin. Erityisesti kehitetyllä tuoterungolla pyrittiin asiakasprojektien toteuttamisen nopeuteen. On mahdoton sanoa, miten kehitysnopeus vertautuu kilpailijoiden vastaavaan. Todennäköisesti kokeneilla tekijöillä on aikojen saatossa hioutuneet metodit, joiden avulla tuote saadaan asiakkaalle. Kehitetty tuoterunkopohjainen ratkaisu antaa kuitenkin keinoja vastata alalla pitkään toimineiden kilpailijoiden ratkaisuihin. On myös vaikea sanoa, kuinka mukautumiskykyinen suunniteltu tuote on verrattuna kilpailijoiden tuotteisiin, koska kilpailijoiden markkinointimateriaalit ovat sen verran pintapuoliset. Todellisen mukautuvuuden tason selvittäminen vaatisi tarkempaa kilpailijoiden haastattelamista ja tutkimista.

Kehitetyllä ratkaisulla pystytään kuitenkin toteuttamaan alan vallitsevat käytännöt ja perusominaisuudet. Tämä on hyvin tärkeä seikka markkinoille pääsemisessä. Näillä ominaisuuksilla ei kuitenkaan saada aikaan kilpailuetua markkinoille. Tarvitaan kehittyneempiä ominaisuuksia kuin perusominaisuudet. Näin ollen kehitetty tuote ei eroa merkittävästi markkinoilla jo olevista tuotteista. Kehittyneiden ominaisuuksien liittäminen osaksi tuoterunkoarkkitehtuuria on selkeä jatkokehityksen tarve. Luvussa 5.2 esiteltiin suunnitelmissa olevia kehittyneitä säätöratkaisuja, joilla olisi mahdollista saada kilpailuetua markkinoilla.

7 YHTEENVETO

Kiinteistöautomaation tarkoituksena on automatisoida kiinteistö. Pääasiassa sen voidaan sanoa vaikuttavan erityisesti neljään eri asiaan: energian hallintaan, turvallisuuteen, käyttömukavuuteen ja rahallisten menojen pienentämiseen. Kiinteistöautomaatio on laajemman mittaluokan automaatiota kuin kotiautomaatio. Tulevaisuudessa on odotettavissa, että automatisoinnin taso kiinteistöissä – niin uusissa kuin saneeratuissa – tulee nousemaan. Kehitys on kulkemassa kohti niin sanottua älytaloa, joka tarjoaa paljon informaatiota käyttäjälle ja toimii ainoastaan tarpeenmukaisesti.

Kiinteistöautomaation tuoterunkoarkkitehtuurin suunnitteleminen on tuotekehitysprosessi. Tällaista prosessia ei kannata aloittaa ilman suunnitelmaa, vaan kannattaa hyödyntää uusien tuotteiden kehittämiseen suunniteltua innovointiprosessimallia. Mallia hyödyntämällä tuotekehitysprosessiin liittyvä riski laskee ja prosessista saadaan paremmin hallittu. Kehitettävä tuote on jatkuvan arvioinnin kohteena ja optimaalista olisi pitää jatkuva kontakti asiakkaaseen. Innovointiprosessimallissa projektin tekeminen jaetaan vaiheisiin ja portteihin. Vaiheet ovat työvaiheita, joissa tapahtuu työn suorittaminen, kuten esitutkimuksen tekeminen ja ohjelmointi. Porttien kohdalla taas pysähdytään arvioimaan miten juuri suoritettu vaihe on onnistunut ja mitä ovat jatkotoimenpiteet.

Tärkeää uuden tuotteen kehittämisessä on ymmärtää millainen on hyvä tuote. Tämän vuoksi tuotekehityksen esitutkimukseen on sijoitettava merkittävästi resursseja. Mitä paremmin vallitsevat markkinat ymmärretään, sitä varmemmin niihin pääsee mukaan. Tärkeää on ymmärtää millainen tuote on sopiva omalle yritykselle. Kiinteistöautomaatioala on erittäin kilpailtu ala, jossa on merkittäviä hinnalla kilpailevia yrityksiä. Yritysten heikkous on se, että ne eivät kykene kunnolla erottumaan kilpailijoista – osittain tästä syystä hintakilpailu on alalla kovaa. Energiatehokkuus ja -hallinta on markkinarako, johon riittävällä osaamisella ja tietoisuudella on mahdollista tunkeutua.

Kiinteistöautomaation ohjelmistot ovat otollisia tuoterunkoarkkitehtuuriin pohjautuvalle ohjelmistotuotannolle, koska niissä on paljon toistuvia rakenteita. Tuoterunkoarkkitehtuuri tarkoittaa ohjelmistoarkkitehtuurityyppejä, jonka on tarkoitus toimia ohjelmistopohjana – tuoterunгон päälle rakennetaan valmiita tuotteita. Tuoterunkoarkkitehtuurin avulla on mahdollista saada merkittäviä säästöjä ohjelmistotuotantoprojekteissa. Ongelmana on tuoterunkoarkkitehtuurin kehittämisen kalleus – kehittäminen vaatii paljon aikaa ja rahaa. Käytännössä kaikissa kiinteistöissä on esimerkiksi ilmanvaihto, lämmönsiirtojärjestelmä ja valojen ohjauksia. Tällaisten pysyvien ominaisuuksien varaan on mahdollista rakentaa tuoterunkoarkkitehtuuri. Tuoterunkoarkkitehtuurilla voidaan myös vastata projektin mukauttamisen haasteisiin. Kun tuoterunгон ominaisuudet toteutetaan modulaarisina komponentteina, joita voidaan tuoterunгон sisällä käyttää

vapaasti, voidaan tarvittavat muutokset toteuttaa näihin komponentteihin ja jättää muu tuoterunko ennalleen.

Projektissa kehitettiin tuoterunkoarkkitehtuuri niin PLC:lle kuin käyttöliittymälle, hyödyntäen kehitystä jaksottavaa innovointiprosessimallia. Tuoterunkoarkkitehtuurit toteutettiin erillisillä ohjelmistoilla. Molemmat tuoterunkoarkkitehtuureihin perustuvat tuoterungot ovat toimintaperiaatteeltaan samanlaisia. Ne koostuvat kerrosarkkitehtuurista ja hyödyntävät modulaarisia black-box-komponentteja, joiden pääasiallinen tarkoitus on vastata tuoterunkoarkkitehtuurien suurimpaan ongelmaan, joka on tuoteperheen eri tuotteiden välinen variaatio.

Innovointiprosessimallissa lähdettiin liikkeelle idean jalostamisesta kohti toimivaa tuotetta ja päädyttiin tuotekehitysvaiheen tuotoksen testaamiseen. Tuloksena kehitetyt tuoterungot vastaavat niille tutkimuksen alussa asetettuja vaatimuksia. Suurimpana ongelmana on kuitenkin testaamisen vajavaisuus johtuen puuttuvasta demoprojektista sekä rajoitetusta simulointiympäristöstä. Tulevaisuuden tehtävät projektissa on saattaa testaaminen sekä innovointiprosessi loppuun, eli viimeistellä tuote.

Tuotteistusprosessin lopullisena tavoitteena on konkretisoida suunniteltu kiinteistöautomaatiojärjestelmän tuoterunkoarkkitehtuuri toimivaksi tuotteeksi. Tavoite saavutetaan viimeistelemällä tuotekehitys ja suorittamalla tuotteistusprosessi loppuun.

8 LÄHDELUETTELO

- [1] Ympäristöministeriö, ”*Rakennuksen automaation vaikutus energiatehokkuuteen*”, 2012. Saatavissa: http://www.avoinautomaatio.fi/doc/standardi_sfs-en_15232/Rakennusten-automaaation-vaikutus-energiatehokkuuteen.pdf. [Haettu 1.3.2016].
- [2] Pakanen, J. ja Lappalainen V., ”*Kotiautomaatio ja koti-ICT*”, Espoo: Valaistuslaboratorio, Raportti 4, 2006.
- [3] Merz H., Hansemann T. ja Hübner C., ”*Building automation*”, Springer Berlin Heidelberg, 2009.
- [4] Kortetjärvi R., ”*Personoiva älykotianturi*”, diplomityö, Tampereen teknillinen yliopisto, 2015.
- [5] Rantala E., ”*Käyttäjälähtöinen älyrakennus*”, [Seminaariesitys]. Yhdyskuntien uudistaminen - seminaari, 2015.
- [6] Sähköinfo, ”*Rakennusautomaatiojärjestelmät - Tietotekniset järjestelmät*”, Sähkötieto ry, 2012.
- [7] Lehtonen T., ”*Taloautomaatio helpottaa elämää ja tuo säästöjä*”, KITA - Kiinteistö & talotekniikka, nro 1, p. 24–28, 2014.
- [8] Cooper R., ”*Winning at new products*”, Basic books, 2011.
- [9] Cooper R., Edgett S., ja Kleinschmidt E., ”*Portfolio management for new product development: Results of an industry practices study*” 2001. Saatavissa: http://www.stage-gate.net/downloads/wp/wp_13.pdf. [Haettu 1 3 2016].
- [10] Cooper R. ja Edgett S., ”*Best practices in the idea-to-Launch process and its governance*” Research Technology Management, vol. 55(2), pp. 43-54, 2012.
- [11] Blackwell G., ”*The Electronic Packaging Handbook*”, CRC Press, 1999.
- [12] IEEE Standardi 1471-2000, ”*IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*”, 2000.
- [13] K. Koskimies ja T. Mikkonen, Ohjelmistoarkkitehtuurit, Talentum, 2005.
- [14] van Vliet H., ”*Software Engineering: Principles*”, Wiley, 2007.
- [15] Hanmer R., ”*Pattern-Oriented Software Architecture for Dummies*”, Chichester: John Wiley & Sons, 2012.
- [16] Harsu M., ”*Ohjelmien ylläpito ja uudistaminen*”, Helsinki, Talentum, 2003.
- [17] Aldekoa, G., Trujillo, S., Sagardui, G. ja Diaz, O., ”*Quantifying Maintainability in*

- Feature Oriented Product Lines*”, *Software Maintenance and Reengineering*, 2008. CSMR 2008. 12th European Conference on Software Maintenance and Reengineering, Ateena, 2008.
- [18] Weiss, D. ja Lai C., ”*Software Product-Line Engineering*”, Addison-Wesley, 1999.
- [19] Bosch J., ”*Design & Use of Software Architectures - Adopting and evolving a product-line approach*”, Addison Wesley, 2000.
- [20] Parviainen R., ”*Tuoterunkoon perustuvan tietovaraston kehittäminen*”, Tampere, Tampereen teknillinen yliopisto, 2014.
- [21] Krueger C., ”*Variation management for software production lines*”, Austin: BigLever Software, Inc, 2002.
- [22] Tiainen T., ”*Haastattelu tietojenkäsittelytieteen tutkimuksessa*”, Tampere, Tampereen yliopisto, 2014.
- [23] Ojala J., ”*Kiinteistöautomaatio järjestelmien käyttäjänäkökulmainen kehittäminen*”, opinnäyte, Haga-Helian ammattikorkeakoulu, 2015.
- [24] Kauppinen J., ”*Ympäristöministeriön asetus rakennusten energiatehokkuuden parantamisesta korjaus- ja muutostöissä*”, Ympäristöministeriö, 2013.
- [25] Mikkola K. ja Riihimäki M., ”*Omakotitalorakentajien valmius ympäristöystävällisiin rakentamistapoihin*”, Espoo, VTT, 2002.
- [26] KNX is Green, ”*Reduced Energy consumption by using home and building control systems*”, [Online]. Saatavissa: <https://www.knx.org/ae/knx/application-areas/knx-is-green/index.php>. [Haettu 11.10.2016].
- [27] Tuomaala P., ”*Yksilöllinen lämpötilansäätö säästää*”, *Automaatioväylä*, osa/vuosik. 32, nro 2, p. 33–34, 2016.
- [28] Mevenkamp M., ”*50 % Energy savins by KNX - details and discussion of a promising result*”, Bremen, Hochschule Bremen, 2007.
- [29] Kurnitski J., ”*Energiatehokas rakennus 2020*”, Rakennusautomaatioseminaari, 2014.
- [30] Marttila S., ”*Home Automation – A Challenge for Electrical Designers, Contractors and Electricians*”, diplomityö, Teknillinen korkeakoulu, 2009.
- [31] Kukkonen P., Hyvärinen J., Saari M. ja Nyman M., ”*Rakennusautomaatio rakentamisen sääntelyssä*”, Espoo: VTT, 2015.
- [32] KvaliMOTV, ”*Mitä laadullinen tutkimus on? Lyhyt oppimäärä.*” [Online]. Saatavissa: http://www.fsd.uta.fi/menetelmaopetus/kvali/L1_2.html. [Haettu 25.1.2017].
- [33] Iloniemi J., *Rakennusautomaatio seminaari*, Seminaariesitys, Beckhoff Automation Oy, 23.2.2016.
- [34] Karjalainen S., ”*Rakennusautomaation käytettävyys*”, Seminaariesitys, VTT, 30.5.2013.

- [35] Koskinen A., ”*Saneeraus uudelle vuosituhannele*” *Automaatioväylä*, osa/vuosik. 32, nro 2, p. 25–26, 2016.
- [36] Techopedia, ”*What does activex control mean?*” [Online]. Saatavissa: <https://www.techopedia.com/definition/270/activex-control>. [Haettu 20.1.2017].

LIITE 1. HAASTATTELUTUTKIMUKSET

1A Yrityksen sisäinen haastattelututkimus

Taulukko: Yrityksen sisäiseen haastattelututkimukseen osallistujat

Henkilö 1.	Suunnittelupäällikkö	Viittauskoodi: a
Henkilö 2.	Energia-alan asiantuntija	Viittauskoodi: b
Henkilö 3.	Rakennussuunnitteluasiantuntija	Viittauskoodi: c

1B Toimialakohtainen haastattelututkimus

Taulukko: Toimialakohtaiseen haastattelututkimukseen osallistuneet

Yritys 1.	Valvomotyöntekijä	Viittauskoodi: a
Yritys 2.	Valvomotyöntekijä	Viittauskoodi: b
Yritys 3.	Huoltomies	Viittauskoodi: c
Yritys 4.	Huoltomies	Viittauskoodi: d
Yritys 5.	Kiinteistöpäällikkö	Viittauskoodi: e
Yritys 6.	Kiinteistön omistaja	Viittauskoodi: f

1C Toimialakohtaisen haastattelututkimuksen apukysymykset

1. Minkä yrityksen suunnittelema järjestelmä rakennuksessa on? Mitä kaikkia järjestelmiä on käytössä?
2. Onko mahdollista arvioida järjestelmän tuottamia säästöjä?
3. Mitkä olivat järjestelmän hankintakriteerit?
4. Jos käytössä on päivitetty versio vanhasta järjestelmästä, mikä on parantunut?
5. Kumpi on tärkeämpää: helppokäyttöisyys vai toimintavarmuus?
6. Minkälaisia toiminnallisuuksia järjestelmässä on?
7. Mitkä ovat tärkeimmät toiminnallisuudet?
 - a. käytettävyyden
 - b. ylläpidettävyyden
 - c. energiatehokkuuden kannalta?
8. Mitä ominaisuuksia nykyiseen järjestelmään olisi hyvä saada lisää?
 - a. Esimerkiksi: siirrettäviä huoneantureita (CO₂, ilmankosteus jne.)?
 - b. Energiamittauksia?
9. Onko järjestelmässä ominaisuuksia, jotka eivät toimi kunnolla tai vaatisivat jatkokehitystä?
10. Miten järjestelmässä on otettu huomioon sen laajentamismahdollisuudet?

11. Tukeeko järjestelmästä matkapuhelin/tabletti-käyttöliittymiä? Olisiko tällainen ominaisuus kiinnostava?
12. Mitä ovat mielipiteesi langattomuudesta: tulevaisuutta, nykyaikaa vai tarpeetonta?

LIITE 2.

Tuoterunkoarkkitehtuurin osajärjestelmien toimintaselostukset

Ilmanvaihtokone

- Käyttö
 - Tuloilmapuhaltimen käyntiä ohjataan aika- ja kalenteriohjauksella.
 - Normaalitilanteessa tuloilmapuhallin on koko ajan päällä.
 - Tuloilmapuhallin ei käy kun:
 - Lämmityspiirin pumppu ei käy (normaalitilanteessa aina päällä jäätyminen estämiseksi)
 - Hätä seis -nappia on painettu
 - Lämmöntalteenottoverkoston pumppua ohjataan tuloilmapuhaltimen käydessä säätöohjelman mukaisesti. Lisäksi pumppu ohjataan säädetyn määrärajan välein päälle hetkeksi jumiutumisen estämiseksi.
 - Poistopuhaltimen käynti on lukittu tuloilmapuhaltimen käyntiin.
 - Tulo- ja poistopuhaltimissa on ohjelmallinen käynnistyshidastus, jolla voidaan määrittää ajanjakso, jossa puhaltimien käynti nousee haluttuun tehoon.
 - Ilmanvaihtokoneessa on kesätuuletuskäyttö. Eli tulopuhallin käy tehostetusti, kun ulkoilman lämpötila ylittää asetetun raja-arvon ja samanaikaisesti poistoilman lämpötila ylittää ulkoilman lämpötilan. Kesätuuletus jatkuu kunnes ulkolämpötila ylittää poistoilman lämpötilan.
- Säädön toiminta
 - Ilmanvaihtokoneen ollessa käynnissä raitis- ja jäteilmapellit ovat auki.
 - Koneen ollessa kiinni, raitis- ja jäteilmapellit ovat kiinni.
 - Lämpötilan säätö:
 - Tuloilman lämpötila pyritään pitämään poistoilman lämpötilaan verrannollisena asetetun lämpötilakäyrän mukaisesti. Lämmitystarpeen kasvaessa ohjataan ensimmäisenä portaana lämmön talteenoton pumppua sekä lämmön talteenottoapiirin pumppua. Toisena portaana ohjataan lämmityspiirin venttiiliä, venttiilille asetetun lämpötilakäyrän mukaisesti.
 - Lämmityspiirin lämpötila pidetään sille asetetussa arvossa lämmitysventtiilin avulla.
 - Lämmön talteenottoapiirin nesteen paluulämpötila ei sallita alittaa asettua raja-arvoa.

- Kanavapaineen säätö:
 - Tulo- ja poistoilmapuhaltimien pyörimisnopeutta säätämällä, kanavapaineet pyritään pitämään niille asetetuissa arvoissa.
 - Mikäli kanavapaine ei saavuta haluttua asetusarvoa tietyssä ajassa, annetaan hälytys.
- Jäätymisvaaratoiminta
 - Kun ilmanvaihtokone on seis pidetään lämmityspiirin lämpötila positiivisena, jäätymisvaaran ehkäisemiseksi.
 - Jos lämmityspiirin lämpötila laskee liian alhaiseksi, tuloilmapuhallin pysähtyy.
- Lämmöntalteenotto
 - Valvotaan lämmöntalteenoton hyötysuhdetta tarvittavien mittausten avulla.
 - Mikäli hyötysuhde tippuu liian alas, annetaan hälytys. Hälytys ei ole voimassa, jos kone on juuri käynnistetty.
- Toimilaitteissa indikointitieto indikointireleen kautta. Ristiriitahälytys jos eri kuin ohjelman

Lämmönsiirtojärjestelmä

- Ohjaukset
 - Lämpimän käyttöveden, lämmitysverkon ja IV-verkoston pumput käyvät jatkuvasti
- Käyttövesiverkon lämpötilan säätö
 - Säädin pitää lämpimän käyttöveden menoveden lämpötilan asetusarvossaan, ohjaamalla siihen kytkettyä säätöventtiiliä
- Lämmitysverkon lämpötilan säätö
 - Säädin pitää lämmitysverkon menoveden lämpötilan asetusarvossaan ohjaamalla siihen kytkettyä säätöventtiiliä.
 - Asetusarvo muuttuu ulkolämpötilan mukaan lämpötilakäyrän mukaisesti.
 - Venttiili ajetaan kiinni, kun tietty ulkolämpötila saavutetaan (kesäsulku)
- IV-verkoston lämpötilan säätö
 - Säädin pitää IV-verkoston menoveden lämpötilan asetusarvossaan ohjaamalla siihen kytkettyä säätöventtiiliä.
 - Venttiili ajetaan kiinni, kun tietty ulkolämpötila saavutetaan (kesäsulku)
- Toimilaitteissa indikointitieto indikointireleen kautta. Ristiriitahälytys jos eri kuin ohjelman

Poistoilmakoje

- Tehtävänä hoitaa tilan poistoilmanvaihto ja yllälämmönpoisto
 - Poistoilmapuhallin on normaalisti seis.

- Tilan lämpötilan noustessa yli asetetun raja-arvon poistoilmapuhallin päälle. Ilmapelti auki. Tämän jälkeen lämmön laskettua riittävästi, puhallin pois päältä ja ilmapelti kiinni.